



# **Universidad Nacional Mayor de San Marcos**

**Universidad del Perú. Decana de América**

**Facultad de Ingeniería de Sistemas e Informática**

**Escuela Profesional de Ingeniería de Sistemas**

## **Implementación de un framework para mejorar la gestión de carga de información al Datawarehouse del Banco Interbank**

### **INFORME DE TRABAJO DE SUFICIENCIA PROFESIONAL**

**Para optar el Título Profesional de Ingeniero de Sistemas**

#### **AUTOR**

**Juan Gabriel SÁNCHEZ IPANAQUÉ**

#### **ASESOR**

**Ciro RODRÍGUEZ RODRÍGUEZ**

**Lima, Perú**

**2018**

## **FICHA CATALOGRÁFICA**

IMPLEMENTACIÓN DE UN FRAMEWORK PARA MEJORAR LA GESTIÓN DE CARGA DE INFORMACIÓN AL DATAWAREHOUSE DEL BANCO INTERBANK

AUTOR: SÁNCHEZ IPANAQUÉ JUAN GABRIEL

ASESOR: RODRIGUEZ RODRIGUEZ CIRO

LIMA – PERU, 2018

TÍTULO PROFESIONAL/GRADO ACADÉMICO: TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

ÁREA/PROGRAMA/LÍNEA DE INVESTIGACIÓN: INGENIERÍAS / TECNOLOGÍA DE INFORMACIÓN Y COMUNICACIÓN / TECNOLOGÍA DE INFORMACIÓN Y APLICACIONES DE SISTEMAS

PREGRADO: UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS – FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA – ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

FORMATO 28 X 20 CM

PÁGINAS: xii, 69

## **DEDICATORIA**

*A mi madre por su apoyo constante e incondicional a lo largo del tiempo, su ejemplo y enseñanzas sobre coraje y perseverancia fueron fundamentales para el logro de este trabajo.*

## **AGRADECIMIENTOS**

Gracias a mi asesor de tesis, el profesor Rodriguez Rodriguez Ciro, por su dedicación y sabios consejos para la culminación de este informe.

A la UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS por haberme acogido entre sus aulas, por haberme permitido adquirir los valiosos conocimientos que me llevaron a terminar este trabajo.

A mis profesores y compañeros del programa por mostrarme que se puede llegar más lejos debatiendo y compartiendo ideas. A mis compañeros con quienes compartí aula durante la carrera, ya que me permitieron conocer otras costumbres, otras realidades, lo cual tuvo un gran impacto en mi crecimiento profesional y personal.

A mi familia por su apoyo y ejemplo, quienes están muy felices de verme culminar esta etapa.

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS  
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**Implementación de un framework para mejorar la gestión de carga de información al  
datawarehouse del banco Interbank**

**Autor:** Sánchez Ipanaqué, Juan Gabriel  
**Asesor:** Rodríguez Rodríguez, Ciro  
**Título:** Informe de Trabajo de Suficiencia Profesional para optar el Título  
Profesional de Ingeniero de Sistemas  
**Fecha:** Diciembre 2018

---

## **RESUMEN**

El presente informe de experiencia profesional describe el Proyecto de Implementación de un Framework para mejorar la gestión de carga de información al datawarehouse del Banco Interbank. El problema que se identificó fue que la empresa no tenía un buen control sobre los distintos procesos que cargaban información al datawarehouse del banco Interbank, por lo que no tenía claro cuántos procesos tenía registrados, cuántos tenían errores o cuántos de ellos ya no estaban siendo ejecutados. Además de ello los desarrolladores no implementaban bajo un marco de trabajo adecuado y se tenían distintos estándares de desarrollo. Ante esto el objetivo planteado fue implementar un marco de trabajo que permita tener control sobre los procesos, así como para que sirva de marco de referencia para los desarrolladores. Para la implementación del framework se utilizó la metodología SCRUM ya que se definieron los módulos requeridos y se planificaron entregables de estos de manera progresiva, lo que permitió recibir feedback y aplicar los ajustes necesarios para los futuros entregables.

**Palabras claves:** Implementación de framework, Datawarehouse, banco Interbank, Unix, carga de información, Teradata, modelo de datos financiero, modelo industria.

**NATIONAL UNIVERSITY OF SAN MARCOS  
FACULTY OF SYSTEMS AND COMPUTER ENGINEERING  
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING**

**Implementation of a framework to improve the management of the information load on  
the Interbank Bank's data warehouse**

**Author:** Sánchez Ipanaqué, Juan Gabriel

**Advisor:** Rodríguez Rodríguez Ciro

**Title:** Professional Sufficiency Work Report for opt for the Professional Title of  
Systems Engineer

**Fecha:** December 2018

---

**ABSTRACT**

This professional experience report describes the Project for the Implementation of a Framework to improve the management of the information load on the Interbank Bank's data warehouse. The problem identified was that the company did not have a good control over the different processes that loaded information into the Interbank bank's data warehouse, so it was not clear how many processes were registered, how many had errors or how many of them were not being executed . In addition, the developers did not implement under an adequate framework and had different development standards. Given this, the objective was to implement a framework that allows control over the processes, as well as to serve as a frame of reference for developers. For the implementation of the framework the SCRUM methodology was used since the required modules were defined and deliverables were planned in a progressive manner, which allowed receiving feedback and applying the necessary adjustments for future deliverables.

**Palabras claves:** Implementation of framework, Datawarehouse, Interbank bank, UNIX, information loading, Teradata, financial data model, industry model.

## ÍNDICE

CARÁTULA EXTERNA .....	i
PÁGINA EN BLANCO .....	ii
CARÁTULA INTERNA.....	iii
FICHA CATALOGRÁFICA .....	iv
DEDICATORIA .....	v
AGRADECIMIENTOS .....	vi
RESUMEN.....	vii
ABSTRACT.....	viii
ÍNDICE .....	ix
ÍNDICE DE FIGURAS .....	xi
ÍNDICE DE TABLAS .....	xii
INTRODUCCIÓN .....	1
CAPÍTULO I - TRAYECTORIA PROFESIONAL .....	3
CAPÍTULO II – CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA .....	6
2.1. EMPRESA – ACTIVIDAD QUE REALIZA .....	6
2.2. VISIÓN .....	6
2.3. MISIÓN.....	6
2.4. ORGANIZACIÓN DE LA EMPRESA .....	7
2.5. ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS .....	7
2.6. EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN.....	7
CAPÍTULO III - ACTIVIDADES DESARROLLADAS.....	9
3.1. SITUACIÓN PROBLEMÁTICA .....	9
3.2. SOLUCIÓN.....	9
3.2.1. <i>Objetivos</i> .....	10
3.2.2. <i>Alcance</i> .....	10
3.2.3. <i>Etapas y metodología</i> .....	10
3.2.4. <i>Fundamentos utilizados</i> .....	14

3.2.4.1.	<i>Programación en Shell script por funciones</i> .....	14
3.2.4.2.	<i>Conexiones a base de datos desde Unix</i> .....	14
3.2.4.3.	<i>Programación de invocación de procesos por parámetros</i> .....	14
3.2.4.4.	<i>Módulos o etapas de procesos</i> .....	15
3.2.4.5.	<i>Arquitectura del framework</i> .....	26
3.2.5.	<i>Implementación de las áreas de procesos y sus buenas prácticas</i> .....	29
3.3.	EVALUACIÓN.....	31
3.3.1.	<i>Evaluación económica</i> .....	31
3.3.2.	<i>Interpretación del van y del tir</i> .....	34
CAPÍTULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA .....		35
4.1.	APORTES EN EL AREA DE DESARROLLO Y RESPONSABILIDADES .....	35
4.1.1.	<i>Puntos de mejora</i> .....	36
4.1.2.	<i>Lecciones aprendidas</i> .....	36
CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES .....		37
5.1.	CONCLUSIONES .....	37
5.2.	RECOMENDACIONES .....	38
FUENTES DE INFORMACIÓN .....		39
GLOSARIO.....		40
ANEXOS - ENTREGABLES DE LA IMPLEMENTACIÓN DEL FRAMEWORK PARA MEJORAR LA CARGA DE INFORMACIÓN AL DATAWAREHOUSE DEL BANCO INTERBANK.....		41



## ÍNDICE DE FIGURAS

Figura 1 – Organigrama del área de Data Analytics .....	8
Figura 2 – Esquema manejo de parámetros en Unix.....	16
Figura 3 – Módulos de carga de información.....	16
Figura 4 – Carga de fuentes de información .....	17
Figura 5 – Carga de información a Landing .....	21
Figura 6 – Carga de información a Staging.....	23
Figura 7 – Carga de información a Datawarehouse .....	25
Figura 8 – Información de Persona .....	26
Figura 9 – Reporte de Cantidad de aperturas de cuentas .....	26
Figura 10 – Arquitectura de carga a Landing.....	27
Figura 11 – Arquitectura de carga a vista de Staging.....	29
Figura 12 – Carga de vista a Layout Staging .....	29
Figura 13 – Arquitectura carga Datawarehouse .....	30
Figura 14 – Configuración cabecera de archivos .....	44
Figura 15 – Configuración detalle de archivos .....	45
Figura 16 – Shell para generar metadata de archivo .....	47
Figura 17 – Shell genérica de carga de archivos .....	48
Figura 18 – Shell genérica de carga de Layout Staging .....	50
Figura 19 – Shell genérica de carga a Datawarehouse .....	51
Figura 20 – Shell para generación de logs de carga .....	56
Figura 21 - Detalle de ejecución de cada proceso .....	57
Figura 22 – Log de error de ejecución .....	57
Figura 23 – Shell de malla de ejecución de procesos.....	62
Figura 24 – Shell de secuencia de ejecución.....	63
Figura 25 - Malla de procesos.....	64
Figura 26 - Reporte de control .....	66
Figura 27 - Reporte de información cargada.....	67
Figura 28 - Reporte de fechas procesadas .....	67

## ÍNDICE DE TABLAS

Tabla 1: Fases y entregables del proyecto .....	12
Tabla 2: Estructura de interface de Canales .....	18
Tabla 3: Lista de entregables.....	30
Tabla 4: Correos de gestión.....	32
Tabla 5: Gastos mensuales por rol .....	32
Tabla 6: Gastos de certificación .....	32
Tabla 7: Gastos de soporte .....	32
Tabla 8: Costo mensual Jobs.....	32
Tabla 9: Ahorros mensuales .....	32
Tabla 10: Flujo de caja durante los 18 próximos meses .....	34
Tabla 11: Descripción campos de tabla de configuración .....	44
Tabla 12: Descripción detalle de tabla de configuración .....	46
Tabla 13: Configuración de malla de procesos. ....	59

## **INTRODUCCIÓN**

El presente informe de experiencia profesional describe el Proyecto de Implementación de un Framework para mejorar la gestión de carga de información al datawarehouse del Banco Interbank.

La empresa Interbank es una de las principales instituciones financieras del país. Tiene una cartera de más de 2 millones de clientes, cuenta con 275 tiendas financieras, más de 1900 cajeros automáticos y 3000 corresponsales Interbank Agente a nivel nacional.

En el año 2017 empezaron a surgir problemas con respecto a los procesos que tiene actualmente el banco en el área de Data Analytics, por lo que se observó la necesidad de contar con un marco de trabajo (Framework), que permita gestionar los procesos de carga de información al datawarehouse, así como para que ayude a los desarrolladores a trabajar sobre estándares de implementación de procesos.

En el CAPÍTULO I se especifican cronológicamente roles y funciones, actividades de la trayectoria profesional.

En el CAPÍTULO II se describe a la empresa donde se desarrolló la experiencia, el banco Interbank, su estructura organizacional, la visión y la misión, los productos y servicios que brinda.

El CAPÍTULO III tiene el detalle del trabajo realizado, el cual fue la implementación de un framework para el área de data analytics.

El CAPÍTULO IV se refiere al aporte del autor, el desarrollo profesional que le demandó, las necesidades que se atendieron, la experiencia y la capacitación requeridas.

En el CAPÍTULO V se detallan las conclusiones y también se indican las recomendaciones.

## CAPÍTULO I - TRAYECTORIA PROFESIONAL

### **PRESENTACIÓN PROFESIONAL**

Bachiller en Ingeniería de Sistemas con experiencia comprobada en implementación de soluciones de Inteligencia de Negocios, Data Warehousing, con conocimientos en análisis, diseño e implementación de aplicaciones Business Intelligence y Data Management para Servicios Financieros, Bancarios, Construcción, Petroleras; mediante el uso de herramientas de Integración de información.

Las empresas en las que el autor ha laborado, los roles y funciones principales se detallan en el siguiente cuadro:

<b>Rol / Empresa</b>	<b>Fecha</b>
Líder técnico Programa de datos  DWCONSULWARE DE PERU - Cliente Interbank - Liderar uno de los equipos de la iniciativa de carga al modelo Industria de Teradata, en el cliente Interbank - Plantear las estrategias y soluciones técnicas, coordinar con usuarios de negocio y otros líderes técnicos sobre el proyecto en curso. - Capacitación y seguimiento a los demás equipos de desarrollo, sobre el framework de carga de información.	01/05/2018 - Actual

<p>Consultor BI</p> <p>DWCONSULWARE DE PERU - Cliente Interbank</p> <ul style="list-style-type: none"> <li>- Relevamiento funcional de procesos de negocio para identificar necesidades, priorizar medidas y definir fuentes de información.</li> <li>- Implementación de Framework para la ingesta al datawarehouse del banco.</li> </ul>	<p>10/07/2017 - 30/04/2018</p>
<p>Consultor BI</p> <p>DWCONSULWARE DE PERU - Cliente Petroperú</p> <ul style="list-style-type: none"> <li>- Implementación de mejoras en el Datamart de Comercial.</li> <li>- Definición de estrategia de pase a Producción</li> </ul>	<p>01/03/2017 - 09/07/2017</p>
<p>Consultor BI</p> <p>DWCONSULWARE DE PERU - Cliente Los Portales</p> <ul style="list-style-type: none"> <li>- Relevamiento funcional de los flujos de procesos del área de viviendas y estacionamientos desarrollados en Qlikview.</li> <li>- Definición de estrategia de extracción y carga a DWH mediante Datastage y Netezza</li> </ul>	<p>01/01/2017 - 28/02/2017</p>
<p>Consultor BI</p> <p>DWCONSULWARE DE PERU - Cliente Interbank</p> <ul style="list-style-type: none"> <li>- Gestión de procesos e identificación de riesgos en conjunto con el equipo Data Analytics del banco.</li> <li>- Relevamiento de información de negocio del área de Canales de venta, solicitudes y adquisiciones.</li> </ul>	<p>01/06/2016 - 31/12/2017</p>
<p>Consultor BI</p> <p>DWCONSULWARE DE PERU - Cliente COFIDE</p> <ul style="list-style-type: none"> <li>- Gestión de procesos e identificación de riesgos en conjunto con el equipo Data Analytics del banco.</li> <li>- Relevamiento de información de negocio del área de Canales de venta, solicitudes y adquisiciones.</li> </ul>	<p>01/04/2016 - 31/05/2016</p>

<p>Consultor BI</p> <p>Gora SAC</p> <ul style="list-style-type: none"> <li>- Definición de KPI's en coordinación con los usuarios líderes de información.</li> <li>- Diseño, transformación y carga de data con IBM Datastage.</li> <li>- Diseño del Modelo Entidad Relación de soluciones BI</li> </ul>	<p>01/08/2014 - 31/03/2016</p>
<p>Analista programador</p> <p>Millenium BI SAC - Cliente Mapfre Perú</p> <ul style="list-style-type: none"> <li>- Resolver incidencias del sistema utilizando PL/SQL.</li> <li>- Responsable de mejoras y correcciones en el módulo de Siniestros, emisión y cobros en coordinación con los usuarios finales.</li> </ul>	<p>01/02/2013 - 31/07/2014</p>

La información académica se detalla a continuación:

<b>Formación</b>	<b>Fecha</b>
<p>Diplomado: Programa de Alta Especialización en Business Intelligence</p> <p>Universidad ESAN.</p>	<p>01/04/2017 – 30/11/2017</p>
<p>Grado Académico de Bachiller en Ingeniería de Sistemas - Escuela Académico Profesional de Ingeniería de Sistemas - Facultad de Ingeniería de Sistemas e Informática - Universidad Nacional Mayor de San Marcos</p>	<p>01/03/2008 – 01/12/2015</p>

## **CAPÍTULO II – CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA**

### **2.1. EMPRESA – ACTIVIDAD QUE REALIZA**

Interbank (Banco Internacional del Perú, S.A.) es una de las principales instituciones financieras del Perú, ya que es una de las que cuenta con mayor participación en el ámbito financiero, con aproximadamente 2 millones de clientes.

Datos de la empresa:

Razón Social: Banco Internacional del Perú, S.A.

Domicilio legal: AV. CARLOS VILLARAN NRO. 140 URB. SANTA CATALINA  
LIMA - LIMA - LA VICTORIA

RUC: 20100053455

Principales proveedores del área DataAnalytics:

- Dwconsultware de Perú.
- Teradata
- Celeritech

### **2.2. VISIÓN**

Ser el mejor banco a partir de las mejores personas

### **2.3. MISIÓN**

Mejorar la calidad de vida de nuestros clientes, brindando un servicio ágil y amigable en todo momento y en todo lugar.



## 2.4. ORGANIZACIÓN DE LA EMPRESA

A continuación se muestra el organigrama de Data Analytics, que es donde se desarrolló la experiencia.

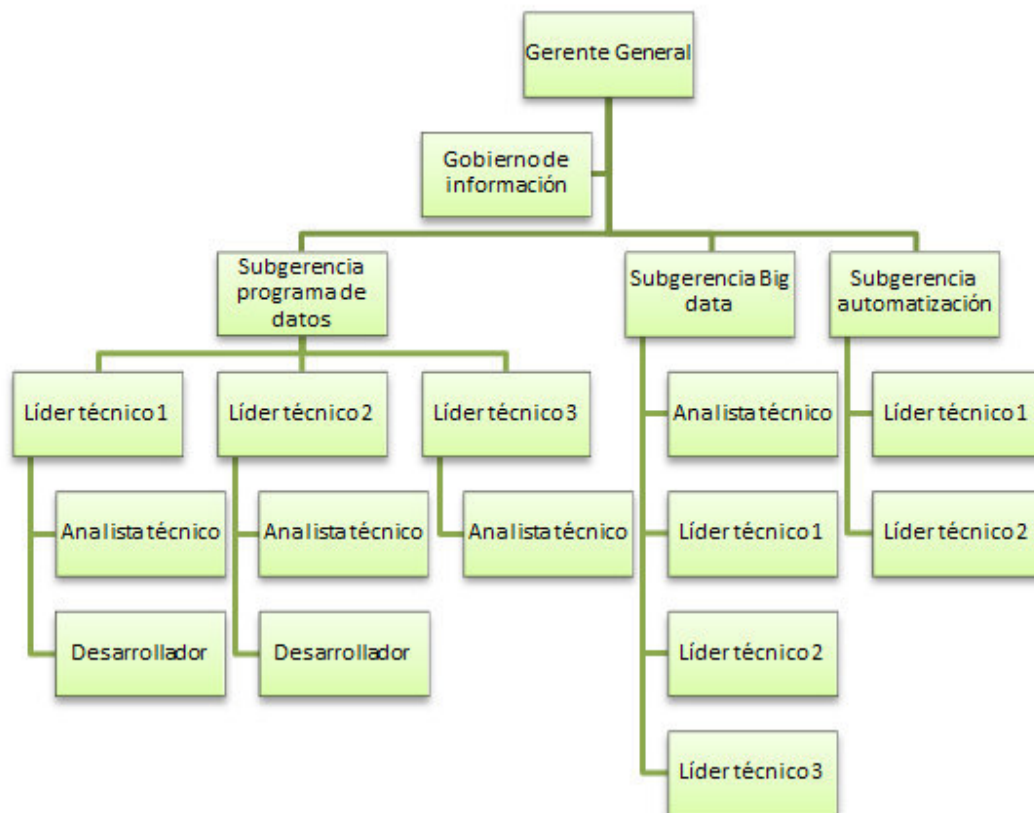


Figura 1 – Organigrama del área de Data Analytics Fuente: Interbank – Data Analytics

## 2.5. ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS

El autor del presente informe se desempeñó como Consultor BI durante la primera fase del proyecto que se expone en este documento, teniendo actualmente el puesto de líder técnico en las nuevas iniciativas que trabajan sobre este marco de trabajo.

## 2.6. EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN

Durante la experiencia profesional en la organización donde se implementó el proyecto, se presentan las funciones realizadas:

- Gestión de procesos e identificación de riesgos en conjunto con el equipo Data Analytics del banco.
- Relevamiento de información de negocio del área de Canales de venta, solicitudes y adquisiciones.
- Implementación de Shell Script, y objetos de BD Teradata para la carga del modelo CRM.
- Definir estrategias de extracción de soluciones BI considerando las mejores prácticas y tiempos de procesamiento con grandes volúmenes de información.
- Liderar uno de los equipos de la iniciativa de carga al modelo Industria de Teradata, en el cliente Interbank.
- Plantear las estrategias y soluciones técnicas, coordinar con usuarios de negocio y otros líderes técnicos sobre el proyecto en curso.
- Capacitación y seguimiento a los demás equipos de desarrollo, sobre el framework de carga de información.

## **CAPÍTULO III - ACTIVIDADES DESARROLLADAS**

### **3.1. SITUACIÓN PROBLEMÁTICA**

El área de operaciones del Banco Interbank tiene inconvenientes para definir un marco de trabajo que permita hacer una ingesta de información, desde las fuentes de información (Interfaces, base de datos) hacia el Datawarehouse, así como identificar problemas con los procesos de carga

#### **CAUSAS**

- No se tiene control sobre los procesos que se ejecutan actualmente.
- En caso se haya caído un proceso (Abortado), no se puede llegar a conocer es el detalle del error.
- No se tiene un estado de ejecución de un proceso de carga de información al warehouse, hora de inicio y fin, así como si finalizó o no de manera correcta.

### **3.2. SOLUCIÓN**

Implementar un framework que permita realizar la ingesta de información al datawarehouse del Banco Interbank, mediante un modelo de permita monitorear los procesos de carga por etapas.

### **3.2.1. Objetivos**

#### **Objetivo general**

Implementar un framework para mejorar la gestión de carga de información al datawarehouse del banco Interbank

#### **Objetivos específicos**

- Incorporar puntos de control de los procesos de carga de información al datawarehouse, por etapas.
- Generar informes que muestren el detalle de los procesos existentes, sus periodos de ejecución y tiempo de procesamiento.
- Tener un marco de trabajo que permita reducir los costos y tiempos de entrega en el desarrollo de procesos de carga al datawarehouse.

### **3.2.2. Alcance**

#### **ALCANCE FUNCIONAL**

El alcance cubre las funciones de monitoreo y control de ejecuciones de los procesos actuales, que se utilizan para la carga de información al Datawarehouse del banco Interbank.

#### **ALCANCE ORGANIZACIONAL**

La Gerencia de Data Analytics, la subgerencia de Operaciones, los cuales se ven beneficiados con la implementación del proyecto.

### **3.2.3. Etapas y metodología**

Para la implementación del presente proyecto se utilizó la metodología SCRUM, ya que se hicieron entregables por etapas.

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado

posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Utilizando esta metodología se dividieron los entregables por fases:

Tabla 1: Fases y entregables del proyecto

Fase	Tarea	Entregable	Técnica empleada
Carga landing	Definición estándares de fuentes de información	- Shells para la carga del archivo a una tabla de Landing, mediante lectura de metadata	1. Reunión con líder técnico para relevar funcionalidades de Carga a Landing
	Registro de metadata del archivo fuente		2. Reunión con Gobierno de información para validar la metadata de los archivos fuente y los estándares a cumplir.
	Carga de la estructura a la tabla de Landing		3. Definición de estructuras que almacenan la metadata, de acuerdo a las funcionalidades requeridas.
Carga Staging	Definición estructura en Staging que almacene lógicas de negocio a	- Shells para lectura de vista y carga de tablas Staging -	1. Reunión con líder técnico para definición de esquemas donde se alojarán las estructuras que contengan las primeras lógicas de negocio (Staging).

	partir de Staging.	Consideraciones de las vistas de negocio en Staging.	2. Definición de consideraciones de las vistas de negocio que almacenan reglas de negocio desde las tablas de Landing.
	Carga de la vista a una estructura que almacene un concepto de negocio (Layout Staging)		3. Pruebas de ejecución desde fuentes, validación de control de errores y feedback del product owner del banco.
Carga Warehouse	Ejecución de un Stored Procedure que lee la tabla de Staging y carga las tablas de Warehouse	- Shells para la ejecución de Stored Procedures y carga de tablas al Warehouse	1. Reunión con el líder técnico para definición de estándares de los stored procedures a implementar.
			- Vistas de reglas de negocio.
			- Manejo de severidades para controlar integridad de información contra lo existente en el DWH.
Control	Registro de las fases de carga en una tabla LOG, donde se ve el detalle de ejecución.	- Shells de Control. - DDL de tablas que almacenan información de las ejecuciones.	1. Reunión con líder técnico para relevar funcionalidades y necesidades de los procesos de control.
			2. Encapsulamiento de funcionalidades en Shell, que permitan invocación de manera genérica.

			3. Programación orientada a la captura de logs y estados de ejecución desde cualquier etapa de la carga de procesos.
Malla de procesos.	Definición de funcionalidad de la malla.	- Objetos de base de datos que almacén información de procesos y grupos. - Shell que controle secuencia de ejecución.	1. Reunión con líder técnico para definir las funcionalidades de la malla a implementar a partir de secuencias de ejecución.
	Definición de objetos de base de datos que almacenen información de procesos.		2. Definición de estructuras que almacenen información de los procesos y grupos de ejecución a los que pertenecen.
	Control de secuencia de ejecución de procesos.		3. Shell que permita controlar la secuencia de ejecución de los procesos implementados.
Reporte de procesos	Generación de reportes de procesos	Dashboard en Power BI, con status de procesos de carga	1. Reunión con líder de explotación sobre estándares de reportes.
			2. Construcción orientada a Dashboard que permitan monitoreo y detección rápida de errores.

Fuente: Elaboración propia

### **3.2.4. Fundamentos utilizados**

#### **3.2.4.1. Programación en Shell script por funciones**

En Shell Script se puede utilizar funciones para agrupar trozos de código de una manera más lógica, esto permite hacer bloques de código reutilizables, lo cual fue útil para la programación Shell Script de este proyecto ya que permitió:

- Generar Shells de Control que puedan ser invocadas desde otros bloques de código, independientemente de la etapa de carga (Landing, Staging, Warehouse), mediante el manejo de parámetros.
- Invocación de las Shell principales que ejecutan una fase de la carga, por cualquier proceso que se hayan registrado
- El proyecto es fácilmente adaptable en otros entornos que requieran la misma funcionalidad, por ejemplo Hadoop.

#### **3.2.4.2. Conexiones a base de datos desde Unix**

- Unix permite configurar conexiones a base de datos mediante archivos de configuración de accesos, lo cual permite desde las Shell invocar parámetros que se hayan registrado en el servidor con los valores de conexión, de manera que no sean accesibles para todas las personas.
- Unix permite intercalar en una misma Shell conexiones a base de datos y manejo de Shell Script, lo cual permite manejar procesos del framework con valores almacenados en Base de datos.

#### **3.2.4.3. Programación de invocación de procesos por parámetros**

Para que el framework sea genérico y pueda ser utilizado por cualquier proceso se utilizó el manejo de parámetros entre Shells, ya que esto permite hacer adaptable el framework a cualquier proceso y módulo de acuerdo a cada etapa de ejecución:



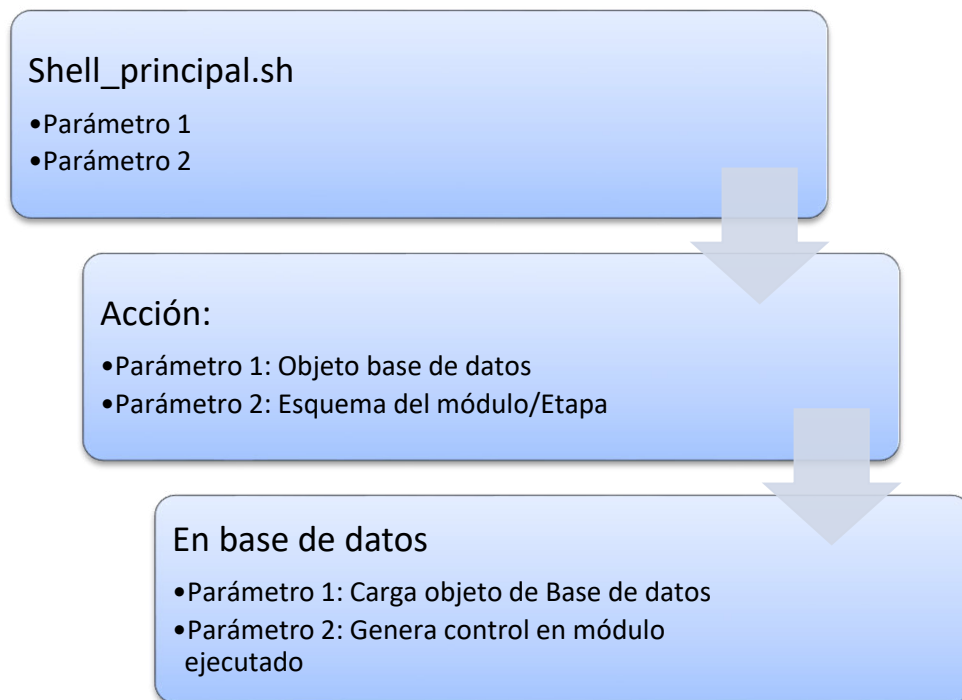


Figura 2 – Esquema manejo de parámetros en Unix Fuente: Elaboración propia

#### 3.2.4.4. Módulos o etapas de procesos

Los módulos por los que fluye la información desde las fuentes hasta el datawarehouse:

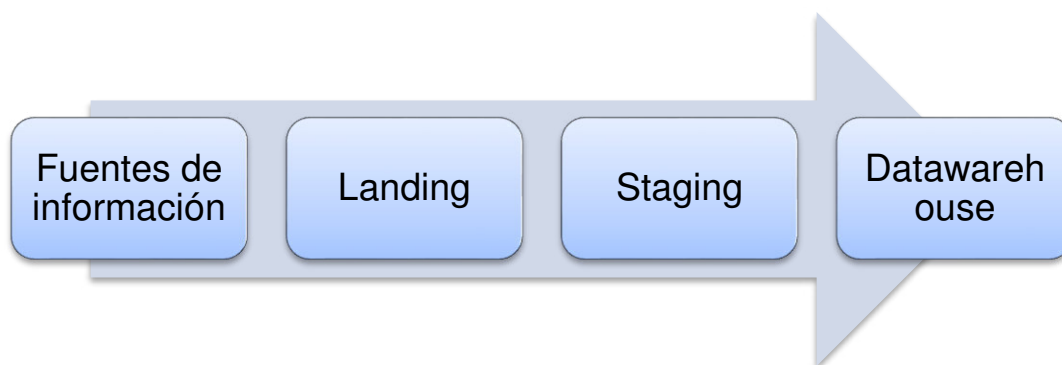


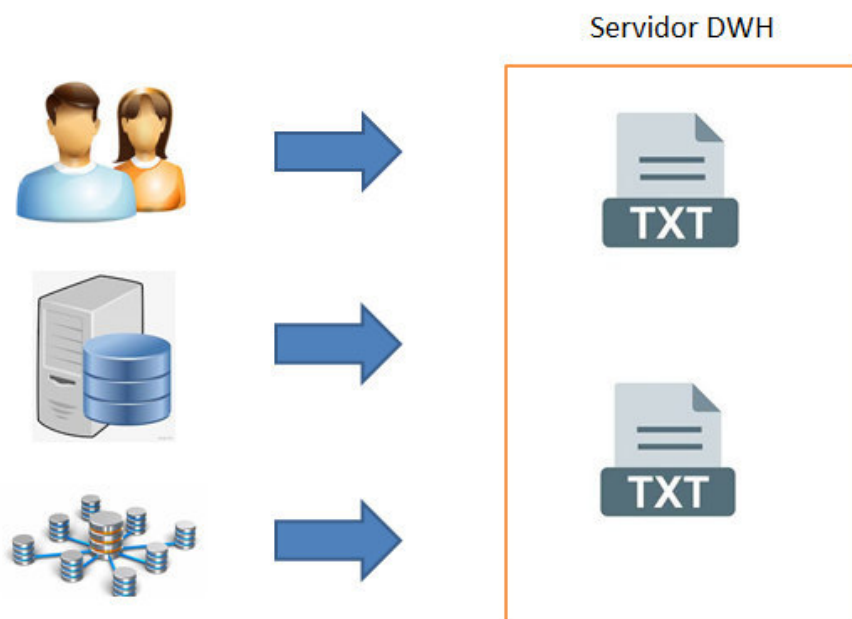
Figura 3 – Módulos de carga de información Fuente: Elaboración propia

A continuación se detalla cada etapa del framework:

### **Fuentes de información**

Esta es la información que tiene la data cruda, puede provenir de:

- Interfaces de usuario.
- Archivos generados desde una base de datos.
- Archivos generados por aplicativos.



**Figura 4 – Carga de fuentes de información      Fuente: Elaboración propia**

Para la generación de archivos se coordina con Gobierno de la información y los usuarios responsables, las consideraciones a tener. Por ejemplo para un archivo (canal\_ATM\_YYYYMMDD.txt):

- Nombre del archivo: canal\_ATM\_YYYYMMDD.txt:
- Se generan en formato txt.
- YYYYMMDD que es parte del nombre, es la fecha en que se genera el archivo (YYYY: Año, MM: Mes, DD: Día).
- La carga es full (Toda la historia)
- La frecuencia de generación será diaria.
- La hora en la que se generarán los archivos será la medianoche (00:00 horas)

- La ruta en la que se generarán será:  
\\grupoib.local\dfs3\DaCRM\09\_Canales\
- Los archivos no deben tener cabecera.
- Los campos están delimitados y el separador de campos debe ser '|'.

Tabla 2: Estructura de interface de Canales:

Número	Campo	Ejemplo
1	IDPTOSERVICIO	IB000268
2	IDCANAL	ATM
3	IDINST	INTB
4	IDLOCACION	223
5	IDHORARIO	1
6	DESCRIPCION	Tienda Pentagonito
7	OBSERVACIONES	
8	FECHAActivacion	00:00.0
9	FECHADESActivacion	00:00.0
10	STATUS	1
11	IDHOPPER	2
12	RCPTPARAM	15032018
13	FOTORUTA	NULL
14	PLANORUTA	NULL
15	COSECHA	4Q 2007
16	PROTOCOLO	TCP/IP
17	TIPORED	Telmex
18	VEL_TX	NULL
19	IP_LOCAL	140.173.3.55
20	MASK_REMOTA	255.255.0.0
21	IP_GATEWAY	140.173.1.1
22	IP_TANDEM	130.30.2.63
23	IP_NATEADA	
24	PUERTO	7777
25	NRO_CELULARCD	
26	IDPSEEBE	466

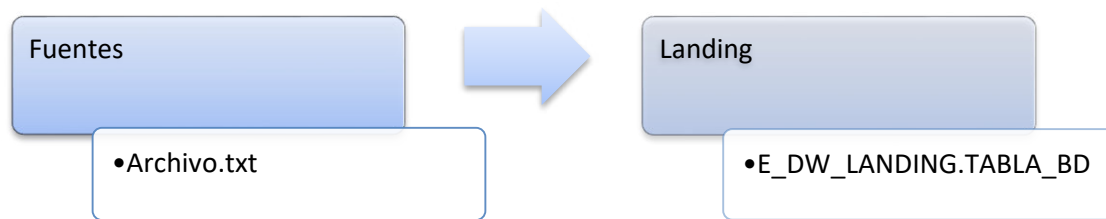
27	FOTORUTAOP	NULL
28	IDSUBCANAL	NULL
29	MUEBLE	No aplica
30	SIRENA	NO
31	INSTALACION	NULL
32	FECHAINSTALACIONOP	NULL
33	IDLOCTIPO	10
34	IDLOCPNTIPO	999
35	LOCACION	Tienda Pentagonito
36	RAZONSOCIAL	Banco Internacional del Perú SAA - INTERBANK
37	RUC	20100053455
38	CODTIENDA	173
39	24HORAS	N
40	DIRECCION	Av. Boulevard Cdra. 9 s/n
41	UBIDIST	150130
42	CONTCOM	
43	CONTOP	
44	TELEFCOM	
45	TELEFOP	
46	ANEXOCOM	617301
47	ANEXOOP	617302
48	FAXCOM	
49	FAXOP	
50	EMAILCOM	ibpentagonito.gte@intercorp.com.pe
51	EMAILOP	ibpentagonito.gte@intercorp.com.pe
52	mostrarBuscador	2
53	COORDSLAT	-12.1008
54	COORDSLON	-76.9883
55	NROSERIE	5300270387
56	IDMODELO	31
57	TRABAS	K - 19
58	TIPOANCLAJE	
59	BOVEDA	UL - 291
60	COD_BARRAS	

61	SIST_OPER	Windows XP
62	IDCONTRATO	34
63	IDLOCPREPROD	0
64	MARCA	Wincor Nixdorf
65	MODELO	Pro Cash 280
66	MODELOTIPO	Lobby
67	TIPOABAST	Frontal
68	PANTALLA	Normal
69	LOCTIPO	Tienda Lima
70	PUNONEUTRO	No aplica
71	HORARIO	Lun a Vie de 8:00 a.m a 5:00 p.m
72	DISTRITO	San Borja
73	PROVINCIA	Lima
74	DEPARTAMENTO	Lima
75	REGION	L
76	zona	Lima Moderna 2
77	IdEjecutivo	b12282
78	Ejecutivo	Claudia Cicirello
79	IdAsistente	b22001
80	Asistente	Karlo Carrillo
81	UBIPROV	1501
82	UBIDEP	15
83	Cod_punoneutro	
84	FECHAPROCESO	20170101
85	USUARIOCREAMODIFI	USUARIO1
86	FECHACREAMODIFI	20170101
87	HORACREAMODIFI	10:01:01

Fuente: Elaboración en conjunto con Gobierno de Información de Data Analytics

## Landing

La información proveniente de los archivos se carga a este ambiente en base de datos.



**Figura 5 – Carga de información a Landing** Fuente: Elaboración propia

Como ejemplo, la estructura que almacena la información del archivo de canales es:

```
CREATE MULTiset TABLE E_DW_LANDING.CRM_SRC_CANAL_ATM ,NO FALLBACK ,  
NO BEFORE JOURNAL,  
NO AFTER JOURNAL,  
CHECKSUM = DEFAULT,  
DEFAULT MERGEBLOCKRATIO  
( idptoservicio VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idcanal VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idinst VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idlocacion VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idhorario VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 descripcion VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 observaciones VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 fechaactivacion VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 fechadesactivacion VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 status VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idhopper VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 rcptparam VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 fotoruta VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 planoruta VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 cosecha VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 protocolo VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 tipored VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 vel_tx VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 ip_local VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 mask_remota VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 ip_gateway VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 ip_tandem VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,
```

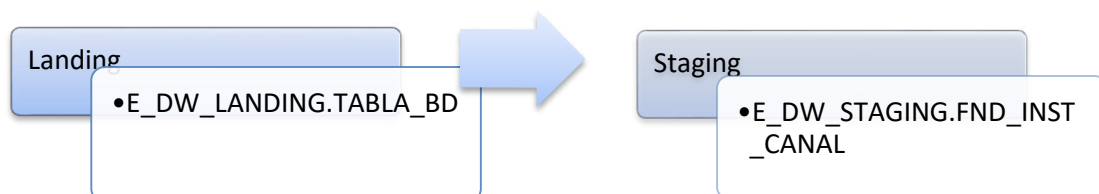
ip\_nateada VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 puerto VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 nro\_celularcd VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idpseebe VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 fotorutaop VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idsubcanal VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 mueble VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 sirena VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 instalacion VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 fechainstalacionop VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idloctipo VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idlocpntipo VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 locacion VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 razonsocial VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 ruc VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 codtienda VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 campo24horas VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 direccion VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 ubidist VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 contcom VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 contop VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 telefcom VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 telefop VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 anexocom VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 anexoop VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 faxcom VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 faxop VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 emailcom VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 emailop VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 mostrarbuscador VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 coordslat VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 coordslon VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 nroserie VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idmodelo VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 trabas VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 tipoanclaje VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 boveda VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 cod\_barras VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 sist\_oper VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idcontrato VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,

idlocreprod VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 marca VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 modelo VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 modelotipo VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 tipoabast VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 pantalla VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 loctipo VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 puntoneutro VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 horario VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 distrito VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 provincia VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 departamento VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 region VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 zona VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idejecutivo VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 ejecutivo VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 idasistente VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 asistente VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 ubiproov VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 ubidep VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 cod\_puntoneutro VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 fechaproceso VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 usuariocreamodifi VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 fechacreamodifi VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC,  
 horacreamodifi VARCHAR(200) CHARACTER SET LATIN NOT CASESPECIFIC)

**PRIMARY INDEX** (idptoservicio);

## Staging

Este ambiente contiene información más estructurada, relacionada a un concepto de negocio, como por ejemplo Canal.



**Figura 6 – Carga de información a Staging**

**Fuente: Elaboración propia**



Ejemplo de una estructura que almacena información de canales:

```
CREATE MULTiset TABLE E_DW_LANDING.FND_INST_CANAL ,NO FALLBACK ,  
NO BEFORE JOURNAL,  
NO AFTER JOURNAL,  
CHECKSUM = DEFAULT,  
DEFAULT MERGEBLOCKRATIO  
  
(  
  
    FechaInformacion DATE FORMAT 'YYYY-MM-DD',  
   CodigoInstanciaCanal VARCHAR(50) CHARACTER SET LATIN NOT CASESPECIFIC,  
    NombreInstanciaCanal VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC,  
    CodigoTipoCanal SMALLINT,  
    CodigoSubTipoInstancialCanal SMALLINT,  
    CodigoEstado SMALLINT,  
    FechaActivacion DATE FORMAT 'YYYY-MM-DD',  
    FechaDesactivacion DATE FORMAT 'YYYY-MM-DD',  
    CodigoTipoRedConexion VARCHAR(50) CHARACTER SET LATIN NOT CASESPECIFIC,  
    CodigoTipoMueble VARCHAR(50) CHARACTER SET LATIN NOT CASESPECIFIC,  
    IndicadorSirena VARCHAR(1) CHARACTER SET LATIN NOT CASESPECIFIC,  
    NumeroSerie VARCHAR(50) CHARACTER SET LATIN NOT CASESPECIFIC,  
    CodigoModeloCanal VARCHAR(50) CHARACTER SET LATIN NOT CASESPECIFIC,  
    CodigoSistemaOperatCanal VARCHAR(50) CHARACTER SET LATIN NOT CASESPECIFIC,  
    CodigoMarcaCanal VARCHAR(50) CHARACTER SET LATIN NOT CASESPECIFIC,  
    UsuarioCreaModificaRegistro VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC,  
    FechaCreacionModifiRegistro DATE FORMAT 'YYYY-MM-DD',  
    HoraCreacionModifiRegistro TIMESTAMP(6))  
  
PRIMARY INDEX ( CodigoInstanciaCanal );
```

## Datawarehouse

Este ambiente contiene estructuras que sirven para el análisis de información, y que se relacionan con los demás objetos de este ambiente de acuerdo a los conceptos asociados.

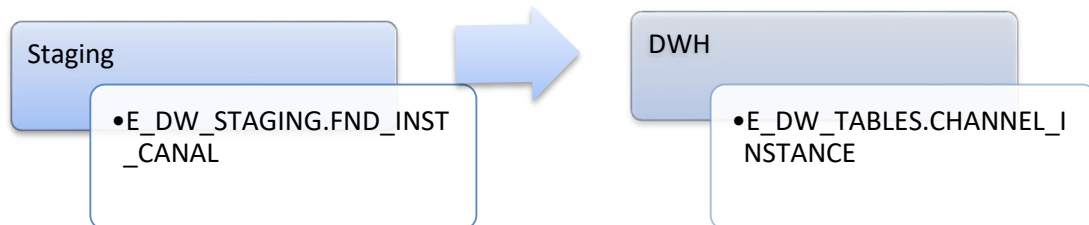


Figura 7 – Carga de información a Datawarehouse Fuente: Elaboración propia

Ejemplo:

Tabla de DWH que almacena información de canales:

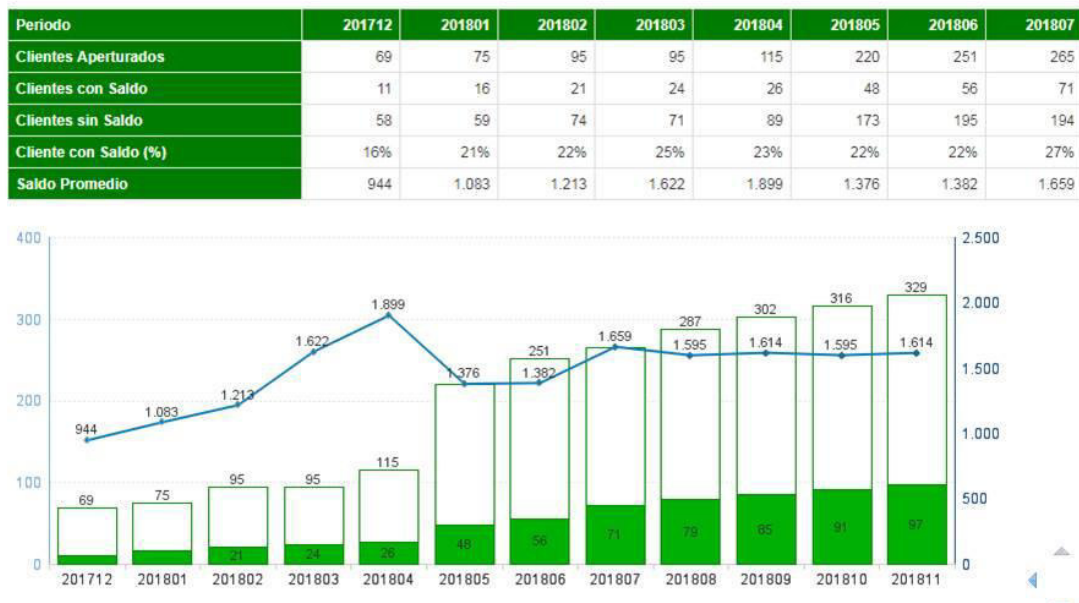
```
CREATE MULTISET TABLE E_DW_TABLES.CHANNEL_INSTANCE ,NO FALLBACK ,
  NO BEFORE JOURNAL,
  NO AFTER JOURNAL,
  CHECKSUM = DEFAULT,
  DEFAULT MERGEBLOCKRATIO
  (Channel_Instance_Id DECIMAL(18,0),
  Channel_Instance_Host_Cd VARCHAR(100),
  Channel_Type_Cd DECIMAL(18,0),
  Channel_Instance_Name VARCHAR(100),
  Channel_Instance_Type_Cd DECIMAL(18,0),
  Channel_Instance_Subtype_Cd DECIMAL(18,0),
  Channel_Instance_Start_Dt DATE FORMAT 'YYYY-MM-DD',
  Channel_Instance_End_Dt DATE FORMAT 'YYYY-MM-DD',
  Convenience_Factor_Cd DECIMAL(18,0),
  Data_Source_Type_Cd DECIMAL(18,0),
  Channel_Status_Cd DECIMAL(18,0),
  Load_Process_Id DECIMAL(18,0),
  Load_Process_Upd_Id DECIMAL(18,0),
  Quality_Type_Cd DECIMAL(18,0)
  )
```

Este módulo permite almacenar información de negocio relacionada a un concepto, por ejemplo, personas:

Party_Id	Party_Identifier_Type_Cd	Party_Identifier_Num	Party_Subtype_Cd	Party_Type_Cd	Gender_Type_Cd	Party_Start_Dt	Party_End_Dt	Birth_Dt	Party_Full_Name
65,156,411	11	0015877594	1	2	?	19/11/2018	31/12/2099	21/02/1951	FELIX MARQUEZ JOSE EDUARDO
62,022,293	11	0015838125	1	2	?	23/10/2018	31/12/2099	04/08/1984	ESCOBEDO MUÑOZ DORIS VELDAD
61,194,564	11	0015637344	11	2	?	18/06/2018	31/12/2099	?	0015637344
62,179,107	11	0015843743	1	2	?	27/10/2018	31/12/2099	11/10/2000	HERRERA FERNANDEZ RENATO ESTEBAN
62,180,045	11	0015844681	1	2	?	29/10/2018	31/12/2099	03/01/1999	MARTICORENA HURTADO BRAYHAN ARMANDO
62,148,254	11	0015823971	1	2	?	15/10/2018	31/12/2099	23/11/1992	TARRILLO HUAMANCHARI EYNER
61,934,832	11	0015826533	1	2	?	16/10/2018	31/12/2099	09/09/1994	MIREZ SOLORZA JOSTEIN ALEXIS
65,105,839	11	0015869432	1	2	?	14/11/2018	31/12/2099	15/05/1999	PEREZ NUÑEZ CARLOS JHONEL
65,058,285	11	0015866492	1	2	?	12/11/2018	31/12/2099	06/12/1997	ASTO AROTINCO MARILU
61,892,845	11	0015807461	1	2	?	03/10/2018	31/12/2099	02/12/1968	VIGO SALAZAR VIVIANA ALCIRA
61,901,899	11	0015816703	1	2	?	10/10/2018	31/12/2099	26/09/1972	SALINAS PECEROS SOLEDAD MARGOT
64,939,971	11	0015852492	1	2	?	02/11/2018	31/12/2099	12/12/1980	INCIARTE GONZALEZ ISRAEL ISAIAS
65,137,630	11	0015873259	1	2	?	16/11/2018	31/12/2099	01/01/1988	SUAREZ MENDOZA RILDO ALAIN
61,898,881	11	0015813602	1	2	?	07/10/2018	31/12/2099	02/09/1990	CHARCA CCOSI EDWIN

**Figura 8 – Información de Persona** Fuente: Data Analytics

Los objetos de este ambiente son los que almacenan la información de negocio y permiten la visualización de reportes, como el siguiente:



**Figura 9 – Reporte de Cantidad de aperturas de cuentas** Fuente: Data Analytics

Para acceder a esta información se utilizan vistas de base de datos, por ejemplo:

```

REPLACE VIEW E_DW_VIEWS.V_DIM_PERSONA_SECTORISTA
AS
LOCKING ROW ACCESS
SELECT
P.Party_Id Party_Id_CUC,
A.Cus_KEY_IB CUC,
A.Officer_CD_IB Sectorista_Cd,
B.Branch_cd Sector_Cd,

```

```

B.Branch_Name SectorDesc
FROM
E_DW_VIEWS.PARTY P
INNER JOIN E_DW_TABLES.CAP_SRC_FEM_PART_DIA A
    ON P.Party_Identification_Num = A.Cus_KEY_IB
    AND P.Party_Type_cd = 2
LEFT JOIN E_DW_TABLES.CRM_SRC_OFSA_ACCOUNT_OFFICER B
    ON A.Officer_CD_IB = B.Account_Officer_Cd

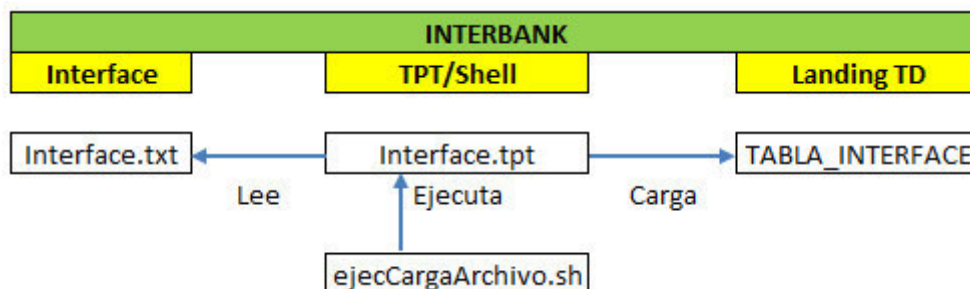
```

### 3.2.4.5. *Arquitectura del framework*

La arquitectura se puede observar por las etapas del proceso:

#### **Landing**

El esquema del funcionamiento de este módulo se muestra en el siguiente diagrama, teniendo como objetivo final, la carga de un archivo a una tabla de Landing:



**Figura 10 – Arquitectura de carga a Landing** Fuente: Elaboración propia

Ejemplo de un archivo de carga tpt (Teradata parallel transport):

```

DEFINE JOB CARGA_TPT_LANDING DESCRIPTION 'CARGA CRM LAYOUT'
( DEFINE SCHEMA TPT_FIX_SCHEMA DESCRIPTION 'SCHEMA DE LA TABLA' (
  CODMES VARCHAR(6),
  CODUNICOCLI VARCHAR(10),
  ADELSUEL VARCHAR(1),
  CONVENIOS VARCHAR(1),
  HIP VARCHAR(1),
  LIBDIS VARCHAR(1),
  TARCRED VARCHAR(1),
  SEGUNDATARCRED VARCHAR(1),
  EXTCASTARCRED VARCHAR(1),
  VEH VARCHAR(1),
  AHORCRED VARCHAR(1),
  CTS VARCHAR(1),
  FFMM VARCHAR(1),
  MILLO VARCHAR(1),
  ALCANCIA VARCHAR(1),

```

```

PLANILLA VARCHAR(1),
PLAZFIJ VARCHAR(1),
TRANSACCIONAL VARCHAR(1),
PA VARCHAR(1),
SEGURO VARCHAR(15),
TENENCIA VARCHAR(15));
DEFINE OPERATOR FILE_READER()
DESCRIPTION 'TERADATA PARALLEL TRANSPORTER DATA CONNECTOR OPERATOR'
TYPE DATACONNECTOR PRODUCER
SCHEMA TPT_FIX_SCHEMA ATTRIBUTES
(VARCHAR DirectoryPath = @Directory,
VARCHAR FileName = @SourceFileName,
VARCHAR IndicatorMode = 'N',
VARCHAR OpenMode = 'Read',
VARCHAR Format = 'DELIMITED',
VARCHAR TextDelimiter= '|',
INTEGER SkipRows = 1);
DEFINE OPERATOR CARGA_DATOS
DESCRIPTION 'TERADATA PARALLEL TRANSPORTER DATA CONNECTOR OPERATOR'
TYPE UPDATE SCHEMA * ATTRIBUTES (
VARCHAR TdpId = @TdpId, VARCHAR UserName= @User,
VARCHAR UserPassword= @Pass,
VARCHAR TargetTable = @TargetDataBase || '.CAP_SRC_TENENCIA_201812',
VARCHAR LogTable = @TargetDataBase || '.CAP_SRC_TENENCIA_201812_LG',
VARCHAR ErrorTable1 = @TargetDataBase || '.CAP_SRC_TENENCIA_201812_ET',
VARCHAR ErrorTable2 = @TargetDataBase || '.CAP_SRC_TENENCIA_201812_UV',
VARCHAR WorkTable = @TargetDataBase || '.CAP_SRC_TENENCIA_201812_WT',
INTEGER MaxSessions = 8 , INTEGER MinSessions = 2 ,
INTEGER TenacitySleep = 1 , VARCHAR DropErrorTable ,
VARCHAR DropLogTable , VARCHAR PauseAcq ,
VARCHAR PrivateLogName ,
VARCHAR ARRAY ErrorList = ['3803','3807','2636']);
STEP CARREGA_TESTE(APPLY
('INSERT INTO '|| @TargetDataBase || '.CAP_SRC_TENENCIA_201812(
CODMES,
CODUNICOCLI,
ADELSUEL,
CONVENIOS,
HIP,
LIBDIS,
TARCRED,
SEGUNDATARCRED,
EXTCASTARCRED,
VEH,
AHORCRED,
CTS,
FFMM,
MILLO,
ALCANCIA,
PLANILLA,
PLAZFIJ,
TRANSACCIONAL,
PA,
SEGURO,
TENENCIA) VALUES (:
CODMES, :
CODUNICOCLI, :
ADELSUEL, :
CONVENIOS, :
HIP, :
LIBDIS, :
TARCRED, :
SEGUNDATARCRED, :
EXTCASTARCRED, :
VEH, :
AHORCRED, :
CTS, :
FFMM, :

```

```

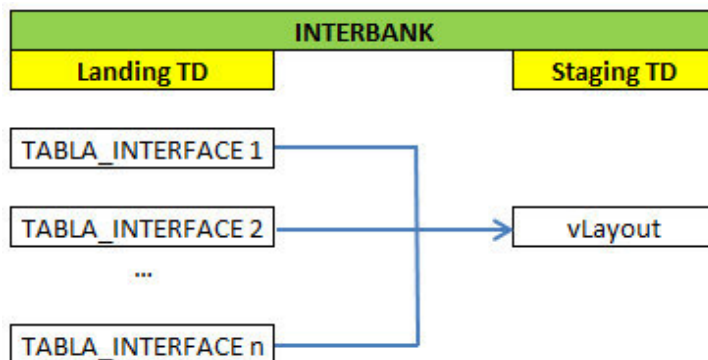
MILLO, :
ALCANCIA, :
PLANILLA, :
PLAZFIJ, :
TRANSACCIONAL, :
PA, :
SEGURO, :
TENENCIA) ;') TO OPERATOR (CARGA_DATOS[1]) SELECT
CODMES,
CODUNICOCLI,
ADELSUEL,
CONVENIOS,
HIP,
LIBDIS,
TARCRED,
SEGUNDATARCRED,
EXTCASTARCRED,
VEH,
AHORCRED,
CTS,
FFMM,
MILLO,
ALCANCIA,
PLANILLA,
PLAZFIJ,
TRANSACCIONAL,
PA,
SEGURO,
TENENCIA FROM OPERATOR(FILE_READER() [1]););

```

## Staging

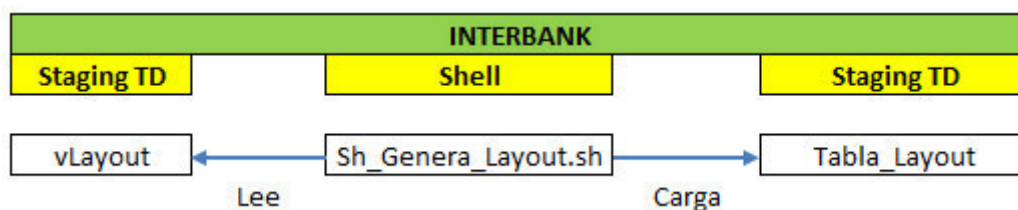
Para este módulo se tienen 2 etapas:

- Carga de la vista que contiene transformaciones a partir de las tablas de landing:



**Figura 11 – Arquitectura de carga a vista de Staging** Fuente: Elaboración propia

- Carga de la vista a una tabla de Base de datos:



**Figura 12 – Carga de vista a Layout Staging** Fuente: Elaboración propia

## Datawarehouse

Este diagrama muestra el flujo de información de carga en DWH, partiendo de las tablas de Staging:

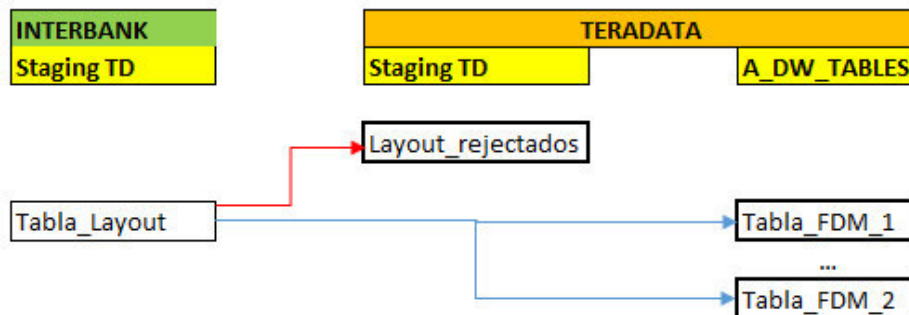


Figura 13 - Arquitectura carga Datawarehouse

### **3.2.5. Implementación de las áreas de procesos y sus buenas prácticas**

Los procedimientos y entregables de acuerdo a la metodología SCRUM, se detallan a continuación, los entregables y los detalles de las funcionalidades se describen en los anexos indicados:

Tabla 3: Lista de entregables.

Procedimiento	Alcance	Anexo	Entregable
Relevamiento de requerimientos	Módulos del framework	Anexo 1	Lista de objetos que componen el Framework
Generación del archivo TPT	Generación de TPT, Objetos de base de datos que almacenen la metadata	Anexo 2	Shell genérica para generación de un archivo TPT
			Objetos de Base de datos que almacenen la metadata de los archivos planos a cargar
Vistas con lógicas de Layout	Definición del estándar de la vista de base de datos para la carga del layout	Anexo 3	Nombre de objeto y definición de estándar para la creación de la vista del layout

Carga de estructuras Layout (Staging)	Carga de tabla en Staging de acuerdo a las reglas de negocio consideradas en la vista de Base de datos	Anexo 4	Shell genérica que lea una vista y cargue una estructura Layout en Staging.
Carga de Datawarehouse	lectura de información de layout para la carga de los objetos del ambiente DWH	Anexo 5	Shell genérica para la ejecución de Stored procedures que carguen el ambiente DWH.
Control por módulos	Objetos de Base de datos para almacenar Controles	Anexo 6	Tablas de base de datos que contengan información de ejecuciones, por módulos.
	Almacenamiento de logs de ejecución de los módulos	Anexo 7	Shell para la carga de las tablas con los estados de ejecución de los distintos procesos.
Malla de procesos	Malla que controle secuencia de ejecución	Anexo 8	Shell para definir la secuencia de ejecución de los procesos relacionados.
Reporte de Operaciones	Reportes de control de procesos	Anexo 9	Dashboard donde se muestre el estado de los procesos ejecutados.

Fuente: Elaboración propia



Tabla 4: Correos de gestión.

Procedimiento	Medio	Anexo	Detalle
Capacitación	Correo electrónico	Anexo 10	Correo de difusión de objetos a los desarrolladores, luego de la capacitación del nuevo framework.

Fuente: Elaboración propia

### 3.3. EVALUACIÓN

#### 3.3.1. Evaluación económica

La evaluación económica incluye:

- Costos del Proyecto
- Personal requerido por mes
- Flujo de caja
- VAN, TIR

#### COSTOS DEL PROYECTO

Los gastos por mes son:

Tabla 5: Gastos mensuales por rol

Personal	Horas por mes	Costo por Hora (S/.)	# de Personas	Costo mensual
Líder técnico	44	60	1	2640
Consultor	176	130	1	22880
Total				<b>25520</b>

Fuente: Elaboración propia

Tabla 6: Gastos de certificación

Evaluación externa	# de Personas	Costo (Mes 6) en S/.
Analista de QA	2	15000

Fuente: Elaboración propia

A partir del mes 7, los gastos mensuales por incidencias del framework y soporte son:

Tabla 7: Gastos de soporte

<b>Personal</b>	<b>Horas por mes</b>	<b>Costo por Hora (S/.)</b>	<b># de Personas</b>	<b>Costo mensual</b>
Consultor	40	130	1	5200
Total				<b>5200</b>

Fuente: Elaboración propia

## **AHORROS ESTIMADOS**

El framework permitirá disminuir la cantidad de Jobs de Control-M, los cuales tienen un costo mensual (S/. 500 soles por job):

Tabla 8: Costo mensual Jobs

<b>Gestión de procesos</b>	<b>Costo mensual</b>
Control-M	2500

Fuente: Elaboración propia

Para el área de operaciones y desarrollo, se espera:

- Prescindir del operador que realizaba las ejecuciones manuales.
- Hacer los desarrollos más eficientes, ya que se tendrá un marco de trabajo referencial, se estima un ahorro de 16 horas de trabajo que realizaban los desarrolladores, en tareas operativas.

Tabla 9: Ahorros mensuales

<b>Personal</b>	<b>Horas por mes</b>	<b>Costo por Hora (S/.)</b>	<b># de Personas</b>	<b>Costo mensual</b>
Operador	176	60	1	10560
Desarrollador	16	100	12	19200
Total				29760
Total (Operaciones + Control-M)				<b>32260</b>

Fuente: Elaboración propia

En base a los gastos y ahorros estimados indicados en las tablas anteriores, se tiene el siguiente flujo de caja, ya que la proyección es que el framework se encuentre vigente durante los 2 próximos años:

Tabla 10: Flujo de caja durante los 18 próximos meses

<b>Mes</b>	<b>Egreso</b>	<b>Egreso</b>	<b>Ingreso</b>	<b>Flujo de caja</b>
<b>201801</b>	-25520			<b>-25520</b>
<b>201802</b>	-25520			<b>-25520</b>
<b>201803</b>	-25520			<b>-25520</b>
<b>201804</b>	-25520			<b>-25520</b>
<b>201805</b>	-25520			<b>-25520</b>
<b>201806</b>	-25520	-15000	32260	<b>-8260</b>
<b>201807</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201808</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201809</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201810</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201811</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201812</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201901</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201902</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201903</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201904</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201905</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201906</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201907</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201908</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201909</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201910</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201911</b>		<b>-5200</b>	32260	<b>27060</b>
<b>201912</b>		<b>-5200</b>	32260	<b>27060</b>

Fuente: Elaboración propia

A partir de estos valores se obtienen el VAN y el TIR:

$VAN = 119391.04$

$TIR = 12\%$

### ***3.3.2. Interpretación del van y del tir***

La empresa Interbank logra un VAN positivo con la implementación del nuevo framework, debido al logro de los siguientes puntos:

- Ahorro de costos en la cantidad de Jobs que se ejecutan actualmente con control-M, se estima 5 jobs menos mensualmente.
- Ahorro en personal de operaciones actual, ya que los procesos se encontrarán centralizados y se reducirá el número de las ejecuciones manuales. Se estima este ahorro en 10560 soles mensuales, lo cual representa un 12% del presupuesto mensual de esta área.
- Se hará más eficiente el trabajo de los desarrolladores, permitiéndoles ahorrar tiempo en tareas operativas, los cuales llevaban en promedio 16 horas mensuales, por los 12 desarrolladores con los que se cuenta actualmente, representa un ahorro de 19200 soles mensuales. Este tiempo se podrá utilizar en implementaciones de otros requerimientos.

Con esto se tiene un VAN de 119391.04, teniendo en cuenta que el framework estará activo los 2 siguientes años.

## **CAPÍTULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA**

### **4.1. APORTES EN EL AREA DE DESARROLLO Y RESPONSABILIDADES**

El autor del presente informe participó en la implementación del framework para mejorar la gestión de carga de información al datawarehouse del banco Interbank, con el rol de Consultor de Inteligencia de negocios.

El equipo además contaba con el líder técnico de Interbank, así como con 2 analistas de calidad, quienes estuvieron apoyando al cierre del desarrollo para revisar las funcionalidades del framework.

El autor también tuvo la responsabilidad de hacer las capacitaciones al personal de operaciones sobre la correcta identificación de los procesos que se ejecutan actualmente, detección de errores y análisis de indicadores con respecto a los procesos, tales como:

- Procesos con mayor cantidad de caídas.
- Procesos registrados sin ejecuciones recientes.
- Procesos que no se ejecutan.

Así como en la capacitación de los desarrolladores, para que puedan utilizar el nuevo framework y sugerir mejoras o hacer un feedback sobre las funcionalidades.

La importancia de la implementación del framework se ve reflejada en la usabilidad que se le está dando para los nuevos procesos de carga al datawarehouse, el análisis de migración de procesos existentes al nuevo framework, esto debido a que le permite al equipo de operaciones monitorear de manera más rápida los procesos existentes, así como conocer el estado de ejecución de cada uno de ellos, para tomar las acciones correctivas que se requieran.

#### **4.1.1. Puntos de mejora**

Durante la implementación del proyecto se identificaron los siguientes puntos de mejora, los cuales se evaluarán para una posterior versión:

1. Para un mayor detalle de las ejecuciones se puede obtener la cantidad de registros procesados en cada módulo del framework y almacenarlos en la misma tabla de Log de ejecuciones.
2. El servidor Unix puede configurarse para el envío de un email al finalizar las ejecuciones.
3. Utilizar una herramienta de reporte que permita publicar los dashboard sin necesidad de instalar la herramienta, para compartir los reportes con links.

#### **4.1.2. Lecciones aprendidas**

Durante la implementación del proyecto se identificaron los siguientes puntos de mejora, los cuales se evaluarán para una posterior versión:

1. Considerar tiempo de estabilización del framework luego de la puesta en producción, ya que se tuvo que hacer algunos ajustes debido al performance cuando ya estaba siendo utilizado por los desarrolladores.
2. Considerar la participación de QA de manera progresiva, ya que realizar la certificación de todos los componentes al final afecta las fechas de puesta en producción.

## **CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES**

### **5.1. CONCLUSIONES**

1. La implementación de los puntos de control del framework permite al área de Data Analytics visualizar de manera fácil y rápida la siguiente información, ya sea por base de datos o mediante reportes:
  - a. Hora de inicio y fin de los procesos.
  - b. Estado de ejecución y en caso de caídas de proceso, cual es el detalle del error.
  - c. Cantidad de procesos existentes, procesos con mayor número de caídas.
2. La información a la que tiene acceso el área de Data Analytics luego de la implementación del framework le permite tomar decisiones correctivas respecto a aquellos procesos que presenten problemas.
3. La implementación y usabilidad del framework por parte de los programadores permite ahorrar 9% de horas-hombre que se utilizaban para tareas de desarrollo, los cuales multiplicado por las 12 personas del equipo representa un ahorro considerable para el banco.
4. La capacitación en el nuevo framework permitió al área de operaciones ahorrar 12% de sus costos mensuales, ya que fue posible prescindir de personal que se requería para tareas operativas.
5. El framework permitirá al datawarehouse de banco consolidarse como el repositorio centralizado de información del banco, ya que el control de procesos se hará de manera más eficiente y ordenada.

## **5.2. RECOMENDACIONES**

1. La gerencia debe garantizar la continuidad del framework, ya que se ha probado con desarrollos de procesos de carga a DWH y sus funcionalidades existentes son las esperadas.
2. Se debe trabajar en nuevas funcionalidades que se requieran, para ello se debe obtener el feedback de los desarrolladores para identificar puntos de mejora.
3. Se debe capacitar y mostrar las funcionalidades del framework a otras áreas relacionadas, para que puedan identificar la forma de utilizar algunas funcionalidades en sus procesos y así poder tener un control sobre todos los procesos de la gerencia de Sistemas de una forma integral.



## **FUENTES DE INFORMACIÓN**

- Francisco Manuel Márquez García. (2004). Unix: programación avanzada. España: RA-MA EDITORIAL.
- Gunther Verheyen. (2013). Scrum – A Pocket Guide. Países Bajos: Van Haren Publishing.
- Classora Knowledge Base. (2018). Arquitectura de una solución de Business Intelligence. 2018, de sinnexus Sitio web:  
[https://www.sinnexus.com/business\\_intelligence/arquitectura.aspx](https://www.sinnexus.com/business_intelligence/arquitectura.aspx)

## **GLOSARIO**

- **INGENIERÍA DE SISTEMAS:**

Es el conjunto de recursos humanos y materiales a través de los cuales se recolectan, almacenan, recuperan, procesan y comunican datos e información con el objetivo de lograr una gestión eficiente de las operaciones de una organización.

- **INTERFACE:**

Archivo que sirve como fuente de información

- **FDM**

Financial data model (Modelo de datos financiero).

- **Datawarehouse**

Un data warehouse es un repositorio unificado para todos los datos que recogen los diversos sistemas de una empresa. El repositorio puede ser físico o lógico y hace hincapié en la captura de datos de diversas fuentes sobre todo para fines analíticos y de acceso.

- **Framework**

Un framework, entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

**ANEXOS - ENTREGABLES DE LA IMPLEMENTACIÓN DEL  
FRAMEWORK PARA MEJORAR LA CARGA DE  
INFORMACIÓN AL DATAWAREHOUSE DEL BANCO  
INTERBANK**

## Anexo 1. Relevamiento de requerimientos

El relevamiento se hizo con el siguiente equipo de Trabajo:

Persona	Rol	Empresa
Eliezer Pachas Quiroz	Líder técnico	Interbank
Juan Sánchez Ipanaqué	Consultor BI	DWConsultware

La lista que se obtuvo luego del relevamiento de las funcionalidades requeridas para el framework es la siguiente:

- ✓ Shell genérica para la generación de un archivo TPT (Teradata parallel transport), el cual permitirá la carga de archivos planos.
- ✓ Shell genérica para la carga de archivos a una tabla de BD (Esquema Landing), invocando al archivo TPT.
- ✓ Definición de vistas que consuman tablas de landing, y mediante lógicas se carguen a una estructura en Staging.
- ✓ Shell para la carga de vistas a estructuras en Staging.
- ✓ Shell genérica para la ejecución de Stored Procedures que carguen información al Datawarehouse.
- ✓ Shell para el control de ejecuciones, por módulos.
- ✓ Manejo de secuencia de ejecución de procesos, mediante tablas que almacenen esta información y Shell genéricas de ejecución.
- ✓ Reportes en Power BI para poder visualizar los procesos existentes, frecuencia y estados de ejecución, para poder tomar las acciones correctivas correspondientes.

## Anexo 2. Generación del archivo TPT

- Para la generación de la tpt se utilizan 2 tablas de configuración que almacenan la información de la metadata de los archivos a cargar.
- a) La tabla CRM\_ESTRUCTURA\_FUENTE, tiene información genérica de las interfaces.
- Cantidad de separadores.
  - Archivo es posición (longitud fija) o por separador (En este caso se debe registrar el carácter separador.
  - El archivo tiene o no cabecera, etc.

**SELECT \* FROM D\_DW\_CONTROL.CRM\_ESTRUCTURA\_FUENTE;**

Answerset 1

	Id_Fuente	nombre_fuente	nombre_tabla_lan	num_separadores	IndLongitudFija	IndSeparador	IndCabecera	SeparadorCampo	FormatoFuente	IndActivo
1	1	T_USO_SBL	CRM_SRC_T_USO_SBL	7	N	Y	Y		bxt	Y
2	2	canal_ATM	CRM_SRC_CANAL_ATM	86	N	Y	Y		bxt	Y
3	3	stocklineas	CRM_SRC_STOCK_LINEAS	68	N	Y	Y	:	bxt	Y
4	4	LIC_Ingresos	CRM_SRC_LIC_INGRESOS	11	N	Y	Y		bxt	Y
5	7	FFLL_Tiendas	CRM_SRC_FFLL_TIENDAS	42	N	Y	Y		bxt	Y
6	8	CANCEMES	CRM_SRC_CANCEMES	?	S	N	N	?	bxt	Y
7	9	Lic_movimientos_comercial	CRM_SRC_LIC_MOV_COMERCIAL	?	S	N	N	?	bxt	Y
8	10	Lic-anulaciones	CRM_SRC_LIC_ANULACIONES	?	S	N	N	?	bxt	Y
9	11	POSCLI	CRM_SRC_POSCLI	?	S	N	N	?	bxt	Y

**Figura 14 – Configuración cabecera de archivos**

**Fuente: Elaboración propia**

Esta tabla debe tener un único registro por cada interface a cargar, para cada campo tener en cuenta:

**Tabla 11: Descripción campos de tabla de configuración**

Campo	Consideración	Ejemplo
Id_Fuente	Identificador de la fuente (Se registra manualmente)	3
nombre_fuente	Es el nombre del archivo, sin la extensión ni la fecha (En caso tenga)	stocklineas
nombre_tabla_lan	Nombre de la tabla de Teradata a la que se carga el archivo	CRM_SRC_STOCK_LI NEAS
num_separadores	Cantidad de caracteres separadores	68



- Esta tabla debe tener un registro por campo de la interface:

Tabla 12: Descripción detalle de tabla de configuración

Campo	Consideración	Ejemplo
Id_Fuente	Este ID debe coincidir con el de la tabla CRM_ESTRUCTURA_FUENTE	3
NombreCampoTecnico	Nombre del campo en la tabla	CodigoUnicoConvenio
IndNulo	Indica si el campo tiene posibilidad de nulo	Y
IndLLavePrimaria	Si el campo es llave primaria	N
IndIndicePrimario	Si el campo es PRIMARY INDEX	Y
Orden	Orden del campo en el archivo	1
TipoCampo	Si es por separador debe ser <b>VARCHAR</b> , si es por posición fija es <b>char</b>	VARCHAR
Longitud	Longitud del campo	200
Escala	Escala de la longitud, en caso se requiera, por defecto 0	0
IndActivo	Si el campo está activo	Y
DescCampo	Descripción del campo	
Ejemplo	Ejemplo de valor del campo	
ValorPorDefecto	Valor que se tenga por defecto	
IndUltimoCampo	Si el campo es el último del archivo	N

Fuente: Elaboración propia

Tener consideración con los siguientes campos: Id\_Fuente, NombreCampoTecnico, Orden, TipoCampo, Longitud, IndUltimoCampo ya que son los más relevantes para la generación de la TPT.

Ejemplo query insert:

```
INSERT INTO D_DW_CONTROL.CRM_CAMPOS_FUENTE (Id_Fuente, NombreCampoTecnico,
IndNulo, IndLLavePrimaria, IndIndicePrimario, Orden, TipoCampo, Longitud, Escala, IndActivo,
DescCampo, Ejemplo, ValorPorDefecto, IndUltimoCampo)
VALUES
(3,'CodigoUnicoConvenio', 'Y','N','Y', 1,'VARCHAR',200, 0, 'Y', NULL, NULL, NULL,'N');
```

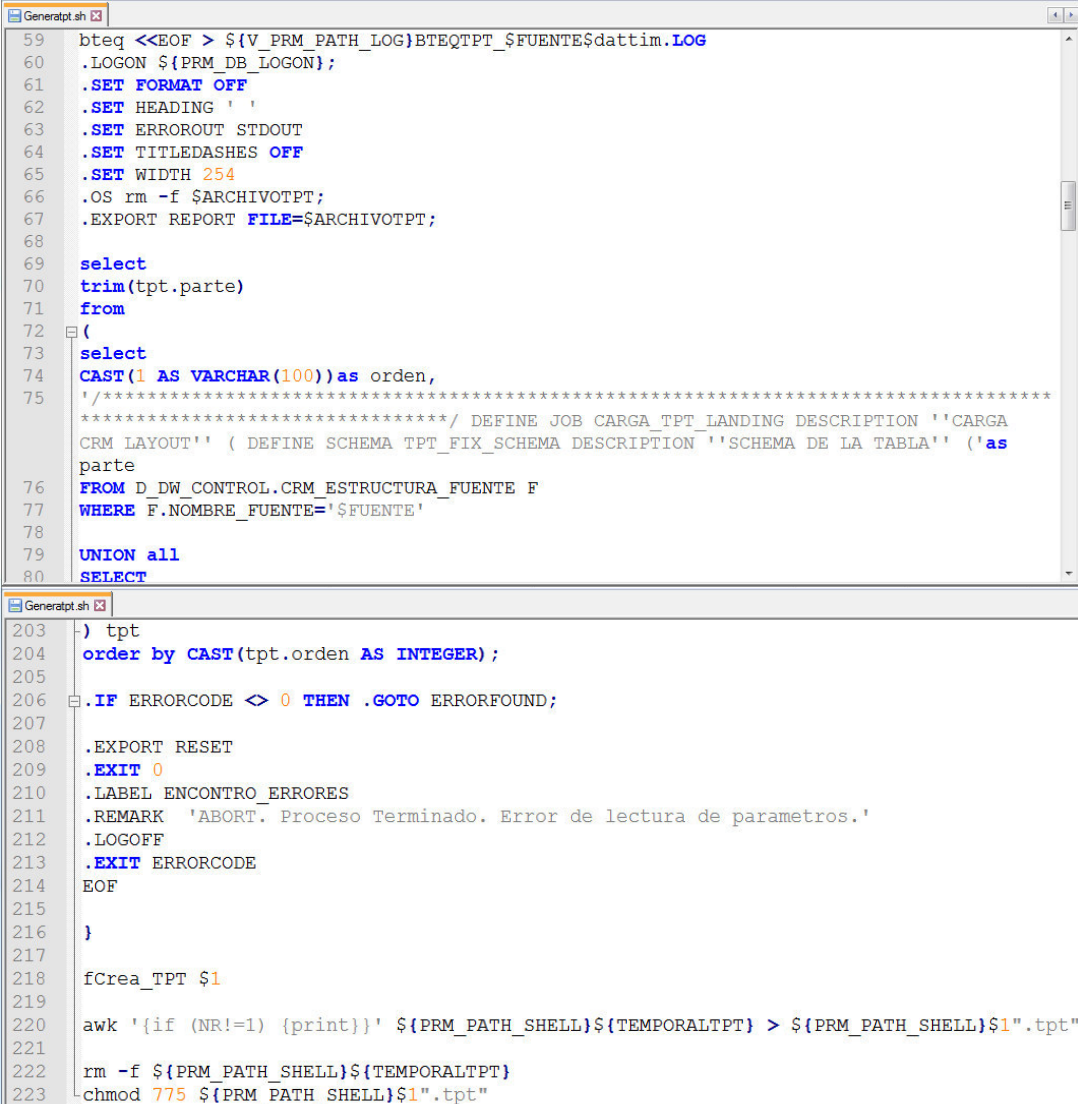
- Se implementó la Shell Generatpt.sh la cual recibe como parámetro el nombre de la interface, la que se ha registrado como nombre\_fuente en la tabla

CRM\_ESTRUCTURA\_FUENTE. Así, para generar el archivo tpt stocklineas.txt y su posterior carga a Teradata se debe ejecutar:

/Staging/CRM/Shell/Generatpt.sh stocklineas

La tpt se generará en la ruta, con el mismo nombre de la interface (Ejemplo: stocklineas.tpt):

/Staging/CRM/Shell/



```

59  bteq <<EOF > ${V_PRM_PATH_LOG}BTEQTPT_${FUENTE}$dattim.LOG
60  .LOGON ${PRM_DB_LOGON};
61  .SET FORMAT OFF
62  .SET HEADING ' '
63  .SET ERROROUT STDOUT
64  .SET TITLEDASHES OFF
65  .SET WIDTH 254
66  .OS rm -f $ARCHIVOTPT;
67  .EXPORT REPORT FILE=$ARCHIVOTPT;
68
69  select
70  trim(tpt.parte)
71  from
72  (
73  select
74  CAST(1 AS VARCHAR(100)) as orden,
75  /*
76  CRM LAYOUT' ( DEFINE SCHEMA TPT_FIX_SCHEMA DESCRIPTION 'SCHEMA DE LA TABLA' ('as
77  parte
78  FROM D_DW_CONTROL.CRM_ESTRUCTURA_FUENTE F
79  WHERE F.NOMBRE_FUENTE='${FUENTE}'
80
81  UNION all
82  SELECT
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103  ) tpt
104  order by CAST(tpt.orden AS INTEGER);
105
106  .IF ERRORCODE <> 0 THEN .GOTO ERRORFOUND;
107
108  .EXPORT RESET
109  .EXIT 0
110  .LABEL ENCONTRO_ERRORES
111  .REMARK 'ABORT. Proceso Terminado. Error de lectura de parametros.'
112  .LOGOFF
113  .EXIT ERRORCODE
114  EOF
115
116  }
117
118  fCrea_TPT $1
119
120  awk '{if (NR!=1) {print}}' ${PRM_PATH_SHELL}${TEMPORALTPT} > ${PRM_PATH_SHELL}${1} ".tpt"
121
122  rm -f ${PRM_PATH_SHELL}${TEMPORALTPT}
123  chmod 775 ${PRM_PATH_SHELL}${1} ".tpt"

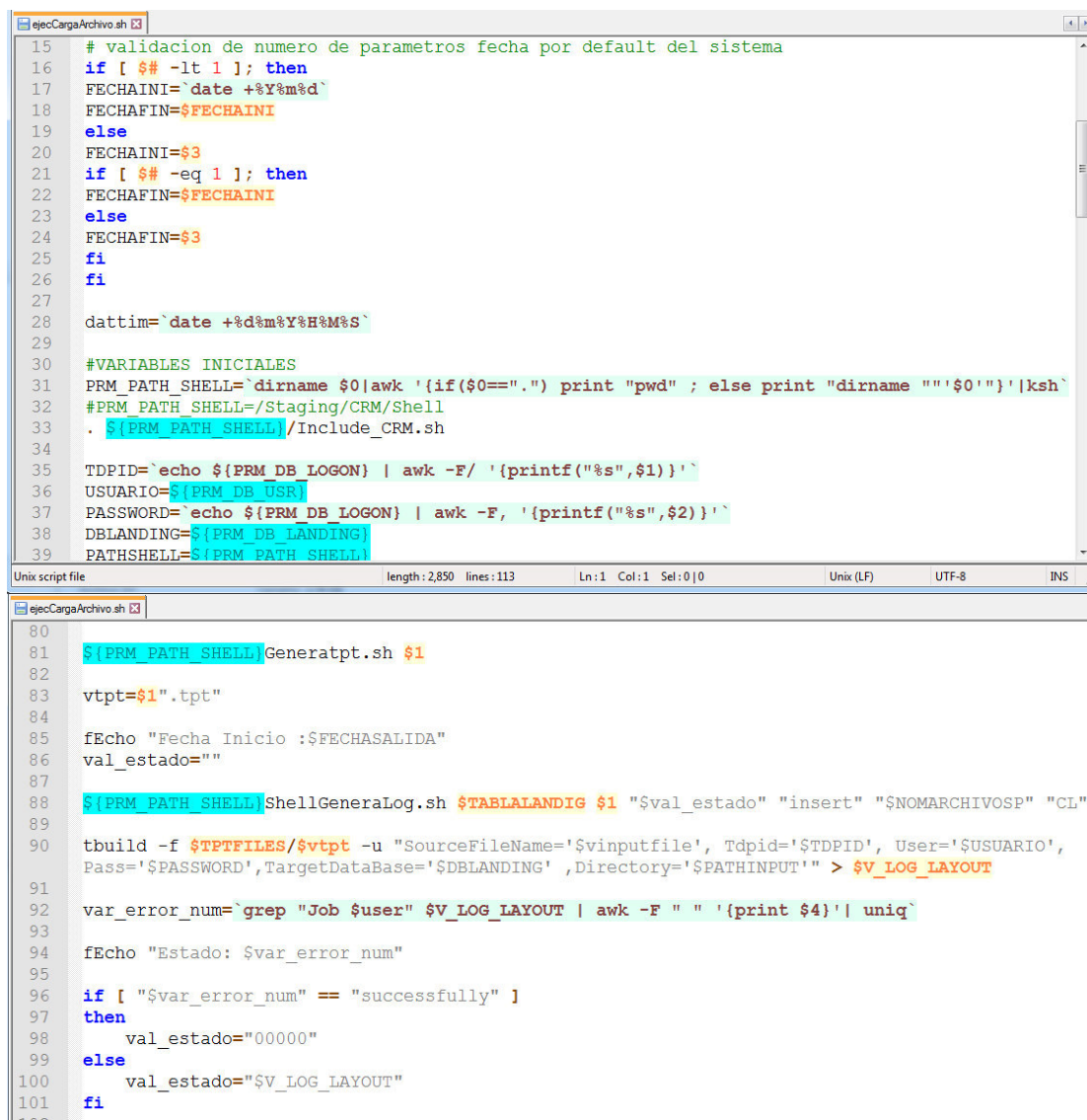
```

Figura 16 – Shell para generar metadata de archivo

Fuente: Elaboración propia



Se implementó la Shell ejecCargaArchivo.sh, la cual invoca a la Shell del punto anterior para cargar la tabla de Landing con la información del archivo fuente:



```

15 # validacion de numero de parametros fecha por default del sistema
16 if [ $# -lt 1 ]; then
17     FECHAINI=`date +%Y%m%d`
18     FECHAFIN=$FECHAINI
19 else
20     FECHAINI=$3
21     if [ $# -eq 1 ]; then
22         FECHAFIN=$FECHAINI
23     else
24         FECHAFIN=$3
25     fi
26 fi
27
28 dattim=`date +%d%m%Y%H%M%S`
29
30 #VARIABLES INICIALES
31 PRM_PATH_SHELL=`dirname $0|awk '{if($0=="") print "pwd" ; else print "dirname "'$0'"'}'|ksh`
32 #PRM_PATH_SHELL=/Staging/CRM/Shell
33 . ${PRM_PATH_SHELL}/Include_CRM.sh
34
35 TDPID=`echo ${PRM_DB_LOGON} | awk -F/ '{printf("%s",$1)}'`
36 USUARIO=${PRM_DB_USR}
37 PASSWORD=`echo ${PRM_DB_LOGON} | awk -F, '{printf("%s",$2)}'`
38 DBLANDING=${PRM_DB_LANDING}
39 PATHSHELL=${PRM_PATH_SHELL}

```

---

```

80
81 ${PRM_PATH_SHELL}Generatpt.sh $1
82
83 vtpt=$1".tpt"
84
85 fEcho "Fecha Inicio :$FECHASALIDA"
86 val_estado=""
87
88 ${PRM_PATH_SHELL}ShellGeneraLog.sh $TABLALANDIG $1 "$val_estado" "insert" "$NOMARCHIVOSP" "CL"
89
90 tbuidl -f $TPTFILES/$vtpt -u "SourceFileName='$vinputfile', Tdpid='$TDPID', User='$USUARIO',
91 Pass='$PASSWORD', TargetDataBase='$DBLANDING' , Directory='$PATHINPUT'" > $V_LOG_LAYOUT
92
93 var_error_num=`grep "Job $user" $V_LOG_LAYOUT | awk -F " " '{print $4}'| uniq`
94
95 fEcho "Estado: $var_error_num"
96
97 if [ "$var_error_num" == "successfully" ]
98 then
99     val_estado="00000"
100 else
101     val_estado="$V_LOG_LAYOUT"
102 fi

```

Figura 17 – Shell genérica de carga de archivos Fuente: Elaboración propia

### Anexo 3. Vistas con lógicas de Layout

Para almacenar las lógicas de carga desde las tablas Landing hacia las estructuras Layout en Staging, estas vistas deben cumplir los siguientes estándares:

- Los campos de la vista deben tener la misma estructura del layout hacia el que se cargarán, en cantidad y tipo de dato

- El nombre de la vista debe ser el nombre del layout de Staging hacia el que se cargarán con el prefijo v: FND\_INST\_CANAL → vFND\_INST\_CANAL.
- El esquema donde se cree la vista debe ser la misma del layout (Staging)

Ejemplo de una vista:

```
create view E_DW_STAGING.vFND_INST_CANAL
as
SELECT
CURRENT_DATE AS FECHAINFORMACION,
IDPTOSERVICIO AS CODIGOINSTANCIACANAL,
TRIM(DESCRIPCION) AS NOMBREINSTANCIACANAL,
10 AS CODIGOTIPOCANAL
-2 AS CODIGOSUBTIPOINSTANCIACANAL,
STATUS AS CODIGOESTADO,
(CASE WHEN UPPER(TRIM(FECHAActivacion))='NULL' THEN NULL
      ELSE TO_DATE(SUBSTR(FECHAActivacion,1,10),'YYYY-MM-DD') END) AS
FECHAActivacion,
(CASE WHEN UPPER(TRIM(FECHAdeActivacion))='NULL' THEN NULL
      ELSE TO_DATE(SUBSTR(FECHAdeActivacion,1,10),'YYYY-MM-DD') END) AS
FECHAdeActivacion,
(CASE WHEN UPPER(TRIM(TIPORED))='NULL' THEN "
      ELSE UPPER(TRIM(TIPORED)) END) AS CODIGOTIPOREDCONEXION,
(CASE WHEN UPPER(TRIM(MUEBLE))='NULL' THEN "
      ELSE UPPER(TRIM(MUEBLE)) END) AS CODIGOTIPOMUEBLE,
(CASE UPPER(TRIM(SIRENA))
WHEN 'SI' THEN 'S'
WHEN 'NO' THEN 'N'
ELSE " END
) AS INDICADORSIRENA,
NROSERIE AS NUMEROSERIE,
(CASE WHEN UPPER(TRIM(MODELO))='NULL' THEN "
      ELSE UPPER(TRIM(MODELO)) END) AS CODIGOMODELOCANAL,
(CASE WHEN UPPER(TRIM(SIST_OPER))='NULL' THEN "
      ELSE UPPER(TRIM(SIST_OPER)) END) AS CODIGOSISTEMAOPERATCANAL,
(CASE WHEN UPPER(TRIM(MARCA))='NULL' THEN "
      ELSE UPPER(TRIM(MARCA)) END) AS CODIGOMARCACANAL,
'USUARIOI' AS USUARIOCREAMODIFICAREGISTRO,
```

```

CURRENT_DATE AS FECHACREACIONMODIFIREGISTRO,
CURRENT_TIMESTAMP AS HORACREACIONMODIFIREGISTRO
FROM E_DW_LANDING.CRM_SRC_CANAL_KIO

```

#### Anexo 4. Carga de estructuras Layout (Staging)

Para la carga de la vista hacia la estructura layout en Staging se utiliza la Shell genérica Sh\_Genera\_Layout.sh:

```

32 .LOGON ${PRM_DB_LOGON};
33 .SET FORMAT OFF
34 .SET HEADING ' '
35 .SET ERROROUT STDOUT
36 .SET TITLEDASHES OFF
37 .SET WIDTH 2000
38 .OS rm -f $ARCHIVOESTRUCTURA;
39 .EXPORT REPORT FILE=$ARCHIVOESTRUCTURA;
40
41 SELECT DISTINCT TRIM(TO_CHAR(LOTE)) FROM $DBSTAGING.v${Layout};
42
43 .IF ERRORCODE <> 0 THEN .GOTO ERRORFOUND;
44
45 .EXPORT RESET
46 .EXIT 0
47 .LABEL ENCONTRO_ERRORES
48 .REMARK 'ABORT. Proceso Terminado. Error de lectura de parametros.'
49 .LOGOFF
50 .EXIT ERRORCODE
51 EOF
52 }
53
54 fEjecutaConsolidaLoteFuente()
55 {
56     filename=${PRM_PATH_SHELL}Slavout" final.txt"

```

```

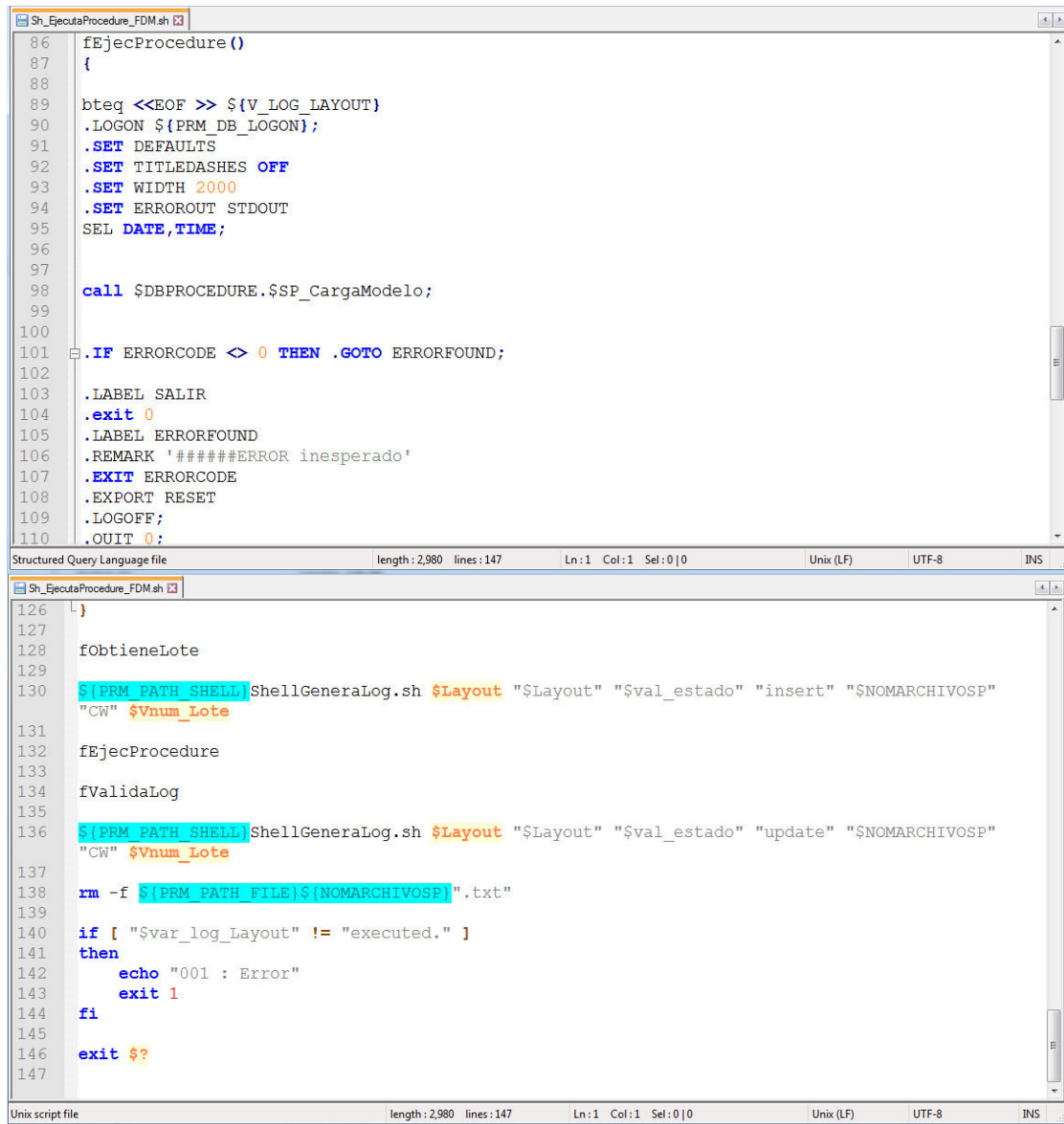
117 fObtieneLote
118 awk '{if (NR!=1) {print}}' ${PRM_PATH_SHELL}${TEMPLOTE} > ${PRM_PATH_SHELL}${Layout}"_final.txt"
119 rm -f ${PRM_PATH_SHELL}${TEMPLOTE}
120 chmod 775 ${PRM_PATH_SHELL}${Layout}"_final.txt"
121
122 fEjecutaConsolidaLoteFuente
123
124 ${PRM_PATH_SHELL}ShellGeneraLog.sh $Layout "$ConsolidaLote" "$sval_estado" "insert"
125 "$NOMARCHIVOSP" "EX"
126
127 fCargaLayout
128
129 fValidaLog
130 ${PRM_PATH_SHELL}ShellGeneraLog.sh $Layout "$ConsolidaLote" "$sval_estado" "update"
131 "$NOMARCHIVOSP" "EX"
132
133 rm -f ${PRM_PATH_FILE}${NOMARCHIVOSP} ".txt"
134
135 if [ "$svar_log_Layout" != "completed." ]
136 then
137     echo "001 : Error"
138     exit 1
139 fi

```

Figura 18 – Shell genérica de carga de Layout Staging Fuente: Elaboración propia

## Anexo 5. Carga de Datawarehouse

Para la carga de las tablas de DWH se utiliza la Shell genérica Sh\_EjecutaProcedure\_FDM.sh, la cual utiliza los procedures que se construyan para la carga de las tablas del FDM (Financial Data Model):



```
86 fEjecProcedure()
87 {
88
89 bteq <<EOF >> ${V_LOG_LAYOUT}
90 .LOGON ${PRM_DB_LOGON};
91 .SET DEFAULTS
92 .SET TITLEDASHES OFF
93 .SET WIDTH 2000
94 .SET ERROROUT STDOUT
95 SEL DATE,TIME;
96
97
98 call $DBPROCEDURE.$SP_CargaModelo;
99
100
101 .IF ERRORCODE <> 0 THEN .GOTO ERRORFOUND;
102
103 .LABEL SALIR
104 .exit 0
105 .LABEL ERRORFOUND
106 .REMARK '#####ERROR inesperado'
107 .EXIT ERRORCODE
108 .EXPORT RESET
109 .LOGOFF;
110 .QUIT 0;
111
112
113
114
115
116 }
117
118 fObtieneLote
119
120 ${PRM_PATH_SHELL}ShellGeneraLog.sh $Layout "$val_estado" "insert" "$NOMARCHIVOSP"
121 "CW" $Vnum_Lote
122
123 fEjecProcedure
124
125 fValidaLog
126
127 ${PRM_PATH_SHELL}ShellGeneraLog.sh $Layout "$val_estado" "update" "$NOMARCHIVOSP"
128 "CW" $Vnum_Lote
129
130 rm -f ${PRM_PATH_FILE}${NOMARCHIVOSP}.txt"
131
132 if [ "$var_log_Layout" != "executed." ]
133 then
134     echo "001 : Error"
135     exit 1
136 fi
137
138 exit $?
```

Figura 19 – Shell genérica de carga a Datawarehouse Fuente: Elaboración propia

Ejemplo de procedure de carga que es ejecutado por la Shell:

```
REPLACE PROCEDURE E_DW_STAGING.SP_PRC_CHANNEL_INSTANCE()
```

```
SQL SECURITY INVOKER
```

```
BEGIN
```

```
--Insert CHANNEL_INSTANCE - Desde Layout histórico --> Después de carga Hist de  
    LYT_CONTRATOS
```

```
create multiset volatile table VOL_CHANNEL_INSTANCE_tmp
```

```
as
```

```
(
```

```
    SELECT A.CodInstCanalVenta, CAST(T.CHANNEL_TYPE_CD AS VARCHAR(50))
```

```
    CodCanalVenta
```

```
    FROM
```

```
    (
```

```
        SEL CodInstCanalVenta,CodCanalVenta
```

```
        FROM E_DW_STAGING.LYT_CONTRATOS
```

```
        GROUP BY 1,2) A
```

```
        LEFT JOIN E_DW_TABLES.CHANNEL_TYPE T
```

```
        ON A.CodCanalVenta=T.CHANNEL_TYPE_HOST_CD
```

```
        LEFT JOIN E_DW_TABLES.CHANNEL_INSTANCE B
```

```
        ON A.CodInstCanalVenta = B.Channel_Instance_Host_Cd
```

```
        and A.CodCanalVenta = T.channel_type_host_cd
```

```
        AND T.CHANNEL_TYPE_CD=B.CHANNEL_TYPE_CD
```

```
        WHERE B.Channel_Instance_Host_Cd IS NULL
```

```
)
```

```
with data
```

```
    primary index(CodInstCanalVenta)
```

```
    on commit preserve rows;
```

```
INSERT          INTO E_DW_TABLES.CHANNEL_INSTANCE
```

```
    (Channel_Instance_Id, Channel_Instance_Host_Cd, Channel_Type_Cd,
```

```
    Channel_Instance_Name, Channel_Instance_Type_Cd, Channel_Instance_Subtype_Cd,
```

```
    Channel_Instance_Start_Dt, Channel_Instance_End_Dt, Convenience_Factor_Cd,
```

```
    Data_Source_Type_Cd, Reg_Update_Dttm, Associate_Human_Resource_Num,
```

```
    Channel_Status_Cd, Load_Process_Id, Load_Process_Upd_Id, Quality_Type_Cd)
```

```
select
```

```
    MAXCATALOGO.MaxId+
```

```
    RANK()OVER(ORDER BY a.codInstCanalVenta ASC) Channel_Instance_Id,
```

```
    a.codInstCanalVenta Channel_Instance_Host_Cd,
```

```

CAST(a.codCanalVenta AS INTEGER) Channel_Type_Cd,
nvl(s.nombresitio,") Channel_Instance_Name,
NULL AS Channel_Instance_Type_Cd,
6 AS Channel_Instance_Subtype_Cd,
'1900-01-01' AS Channel_Instance_Start_Dt,
'2099-12-31' AS Channel_Instance_End_Dt,
NULL AS Convenience_Factor_Cd,
NULL AS Data_Source_Type_Cd,
NULL AS Reg_Update_Dttm,
NULL AS Associate_Human_Resource_Num,
1 AS Channel_Status_Cd,
--cast(current_date as date format 'YYYYMMDD') AS Load_Process_Id,
(SELECT
TRIM(SUBSTR(TO_CHAR(LAYOUTCD),1,4)||TO_CHAR(CURRENT_TIMESTAMP,'YYYYMM
DDHH24MISS')) FROM E_DW_CONTROL.DBC_LAYOUT
WHERE NombreTecnicoLayout='FND_CHANNEL_INSTANCE') AS
Load_Process_Id, NULL AS Load_Process_Upd_Id,
1 AS Quality_Type_Cd
from
VOL_CHANNEL_INSTANCE_tmp a
left join E_DW_LANDING.FND_SITIO s
on a.codInstCanalVenta=s.codigositio
left join E_DW_TABLES.CHANNEL_INSTANCE i
on a.codInstCanalVenta=i.Channel_Instance_Host_Cd
AND a.CodCanalVenta=i.CHANNEL_TYPE_CD
CROSS JOIN
(SEL MAX(CAST(Channel_Instance_Id AS INTEGER)) AS MaxId
FROM E_DW_TABLES.CHANNEL_INSTANCE) MAXCATALOGO
where i.Channel_Instance_Id is null;

-----FIN PROCEDURE-----
END;
```

## Anexo 6. Objetos de Base de datos para almacenar Controles

Para guardar la información de control se utilizarán las tablas:

**E\_DW\_CONTROL.DBC\_LAYOUT:** En esta tabla se registran los procesos que se hayan implementado y que se ejecuten mediante el Framework.

Registro de un proceso en la tabla:

**INSERT INTO E\_DW\_\_CONTROL.DBC\_LAYOUT**

(LayoutCD, NombreTecnicoLayout, IndValidacionLlavePrimaria, IndPermiteRegistrosDuplicados, TipoCargaCD, DelimitadorCampo, DefinicionCampoNulo, FormatoFecha, DelimitadorTexto, NumeroErroresPermitidos, IndRecolectaEstadisticas, VersionLayout, IndValidaCtl, DDLTablaDestino, PeriodicidadCargaCD, TemaCD, FechaUltModificacion, ElaboradoPor, DescripcionLayout, Nombre, BaseDatosTablaStage, IndActivo, IndCoreGlobal, IndGeneracionLayout, IndConcatenarArchivos, TipoCargaStage, IndGeneraCatalogos, EntidadFuente, CondicionFuente, IndCabeceraLayout, IndDiferencial,CodigoMallaIBK, GrupoMallaIBK, Ind\_Encryptacion, Campo\_Encryptado, IndFechaArchivo)

**VALUES**

(1374001, 'FND\_CATALOGO','N', 'N',  
'1', '1', NULL, 'YYYYMMDD',  
'', 0, 'N',  
'1.0', 1, NULL, '1',  
1, '2018-02-23', 'JSANCHEZI', 'Layout para la carga de catalogos',  
'FND\_CATALOGO', NULL, 'Y', NULL, 'N',  
'N', '1', 'Y', 'FND\_CATALOGO',  
NULL, 'N', 'N', NULL,  
NULL, 'N', NULL, 'Y');

**E\_DW\_CONTROL.DBC\_LOGEJECUTAPROCESOETL:** Almacena la información de las ejecuciones de los procesos, como:

- Nombre del proceso.
- Fecha y hora de Inicio.
- Fecha y hora de fin.
- Estado de la ejecución.
- En caso haya error, donde visualizar el detalle del error.

## Anexo 7. Almacenamiento de logs de ejecución de los módulos

- Se implementó el Stored procedure de Control, el cual en base a los parámetros que recibe ingresa un nuevo registro a la tabla de Log, con una nueva ejecución de un proceso, o actualiza su estado al finalizar la ejecución:

```
REPLACE PROCEDURE A_DW_STAGING.SP_IBK_Control
(IN_Layout VARCHAR(100), IN_Status VARCHAR(100), In_Etapa VARCHAR(30), IN_V_Process_ID
    DECIMAL(18,0), NOM_FUENTE VARCHAR(400), IN_Modulo VARCHAR(20))
SQL SECURITY INVOKER
BEGIN
DECLARE IN_V_User VARCHAR(50);
SELECT USER INTO IN_V_User;
-----
if (In_Etapa='insert') then
INSERT      INTO A_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
(Layout, Modulo, Job, FechaHoraInicio, FechaHoraFin, Paso, Usuario, Estado, EstadoDetalle,
    LayoutInstancia, Lote,
    AmbienteEjec, ProyectoETL, UsuarioEjecETL, TipoEjecucion, FechaData, Reproceso, MallaSessionCd)
VALUES
(IN_Layout, IN_Modulo, NULL, CAST(CAST(CURRENT_TIMESTAMP AS VARCHAR(19)) AS
    TIMESTAMP(0)), NULL, 1, IN_V_User, NULL, 'Carga de:
    '||NOM_FUENTE, NULL, IN_V_Process_ID,
    'A', NULL, NULL, IN_Modulo, TO_CHAR(CURRENT_DATE, 'YYYYMMDD'), NULL, NULL);
else
UPDATE A_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
SET FechaHoraFin = CAST(CAST(CURRENT_TIMESTAMP AS VARCHAR(19)) AS
    TIMESTAMP(0)),
estado=(case trim(IN_Status) when '00000' then 'Ok'
                                when '01000' then 'Ok'
                                when '02000' then 'Ok'
                                when 'successfully' then 'Ok'
                                else 'Error: '||IN_Status end),
EstadoDetalle='Carga de: '||NOM_FUENTE
WHERE Lote=IN_V_Process_ID
AND FechaHoraInicio=(SELECT MAX(FechaHoraInicio) FROM
    A_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL C
WHERE C.Lote=IN_V_Process_ID);
end if;
END;
```



- Se implementó una Shell de control ShellGeneraLog.sh, la cual se invoca antes y después de la ejecución de un proceso, para capturar el inicio y fin, así como el estado de ejecución y almacenarlos en las tablas indicadas en el punto anterior:

```

33 fGenera_query_Insert()
34 {
35     ARCHIVOESTRUCTURA=${PRM_PATH_FILE}${NOMARCHIVO} ".txt"
36
37     bteq <<EOF > ${V_LOG_LAYOUT}
38     .LOGON ${PRM_DB_LOGON};
39     .SET FORMAT OFF
40     .SET HEADING ' '
41     .SET ERROROUT STDOUT
42     .SET TITLEDASHES OFF
43     .SET WIDTH 2000
44     .OS rm -f $ARCHIVOESTRUCTURA;
45     .EXPORT REPORT FILE=$ARCHIVOESTRUCTURA;
46
47     SELECT TRIM(SUBSTR(TO_CHAR(LAYOUTCD),1,4))||TO_CHAR(CURRENT_TIMESTAMP,'YYYYMMDDHH24MISS')||
48     '~'||
49     'SP_IBK_Control('''||NombreTecnicoLayout||'', ''||'$VAR_STATUS' ||'', ''||'$VAR_ETAPA' ||
50     '','' ||
51     TRIM(SUBSTR(TO_CHAR(LAYOUTCD),1,4))||TO_CHAR(CURRENT_TIMESTAMP,'YYYYMMDDHH24MISS') ||'', ''
52     ||'$NOM_FUENTE' ||'', ''||'$VAR_MODULO' ||'')'
53     || '~HEADER' ComandoSP
54     FROM ${PRM_DB_GESTORCONFIG}.DBC_LAYOUT
55     WHERE NombreTecnicoLayout='${PROCESO}';
56
57     IF ERRORCODE <> 0 THEN .GOTO ERRORFOUND;
58 }
59
203 if [ "$VAR_MODULO" != "CW" ]
204 then
205     if [ "$VAR_ETAPA" == "insert" ]
206     then
207         fGenera_query_Insert
208         ${PATHSHELL} Sh_EjecutaProcedure.sh "${VComandoSP}" "${DBSTAGING}"
209     else
210         if [ "$VAR_MODULO" != "CL" ]
211         then
212             fGenera_query_Update
213             fActualizaLote ${DBSTAGING}
214             ${PATHSHELL} Sh_EjecutaProcedure.sh "${VComandoSP}" "${DBSTAGING}"
215         else
216             fGenera_query_Update
217             fActualizaLote ${DBLANDING}
218             ${PATHSHELL} Sh_EjecutaProcedure.sh "${VComandoSP}" "${DBSTAGING}"
219         fi
220     fi
221 else
222     if [ "$VAR_ETAPA" == "insert" ]
223     then
224         fGenera_query_Insert_FDM
225         ${PATHSHELL} Sh_EjecutaProcedure.sh "${VComandoSP_FDM}" "${DBSTAGING}"
226     else
227         fGenera_query_Update_FDM
228         ${PATHSHELL} Sh_EjecutaProcedure.sh "${VComandoSP_FDM}" "${DBSTAGING}"
229     fi
230 fi
231
232

```

Figura 20 – Shell para generación de logs de carga Fuente: Elaboración propia

Esto permite conocer el detalle de ejecución de cada proceso, como se muestra en el siguiente cuadro:

Grupo	Layout	Modulo_Layout	Estado	FechaHoraInicio	FechaHoraFin
Dotacion_CL	CRM_SRC_UNIDAD_ORG	CL	Ok	2018-11-07 10:08:58	2018-11-07 10:09:20
Dotacion_CL	CRM_SRC_ABP_CONTACT	CL	Ok	2018-11-07 10:08:59	2018-11-07 10:09:25
Dotacion_CL	CRM_SRC_FFLL_TIENDAS	CL	Ok	2018-11-07 10:09:00	2018-11-07 10:09:24
Dotacion_CL	CRM_SRC_ADQ_CLIENTES	CL	Ok	2018-11-07 10:09:00	2018-11-07 10:09:26
Dotacion_CL	CRM_SRC_Dotacion_EJECUTIVO	CL	Ok	2018-11-07 10:09:00	2018-11-07 10:09:22
Dotacion_CL	CRM_SRC_DOTACION_FFVV_PLANILLA	CL	Ok	2018-11-07 10:09:00	2018-11-07 10:09:25
Dotacion_CL	CRM_SRC_DOTACION_BCA_TEL	CL	Ok	2018-11-07 10:09:00	2018-11-07 10:09:24
Empleado_EX	FND_REEMCAN	EX	Ok	2018-11-07 10:19:31	2018-11-07 10:19:33
Empleado_EX	FND_EMPLEADO	EX	Ok	2018-11-07 10:19:31	2018-11-07 10:19:34
Contratos_ST_HST	LYT_CONTRATO_SAVING_HIST	CW	Ok	2018-11-07 10:20:21	2018-11-07 10:20:44
Contrato_Saldo_EX	LYT_CONTRATO_SALDO	EX	Ok	2018-11-07 10:21:40	2018-11-07 10:22:16
Contrato_Saldo_EX	LYT_CONTRATOS	EX	Ok	2018-11-07 10:21:40	2018-11-07 10:25:11
Contrato_Saldo_EX	LYT_PERSONA	EX	Ok	2018-11-07 10:21:40	2018-11-07 10:23:25
Canal_Empleado_CW	FND_CHANNEL_INSTANCE	CW	Ok	2018-11-07 10:25:55	2018-11-07 10:25:58
Canal_Empleado_CW	FND_LYT_EMPLEADO	CW	Ok	2018-11-07 10:25:56	2018-11-07 10:26:43
Rel_EmpCanal_CW	FND_RELEMPCANALCAPTAC	CW	Ok	2018-11-07 10:27:41	2018-11-07 10:28:13
Persona_CW	LYT_PERSONA	CW	Ok	2018-11-07 10:28:58	2018-11-07 10:29:17
Contratos_CW	LYT_CONTRATOS	CW	Ok	2018-11-07 10:30:18	2018-11-07 10:41:06
Saldo_CW	LYT_CONTRATO_SALDO	CW	Ok	2018-11-07 10:42:02	2018-11-07 10:46:12
Saldo_Cta_Sueldo_CW	LYT_SALDOS_CTASUELDOIND	CW	Ok	2018-11-07 10:57:19	2018-11-07 10:57:22
Contrato_Cta_Sueldo_CW	FND_CONTRATO_CTASUELDOIND	CW	Ok	2018-11-07 10:58:05	2018-11-07 10:58:14
Contratos_Rel_EX	LYT_CARACTERISTICAS	EX	Ok	2018-11-07 10:58:59	2018-11-07 11:01:45
Contratos_Rel_EX	LYT_REL_CONTRATO_ESTADO	EX	Ok	2018-11-07 10:59:00	2018-11-07 11:03:04
Contratos_Rel_EX	LYT_REL_CONTRATO_PRODUCTO	EX	Ok	2018-11-07 10:59:00	2018-11-07 11:02:00
Contratos_Rel_EX	LYT_REL_PERSONA_CONTRATO	EX	Ok	2018-11-07 10:59:00	2018-11-07 11:00:48
Contratos_Rel_EX	LYT_REL_CONTRATO_CARACTERISTICAS	EX	Ok	2018-11-07 10:59:00	2018-11-07 11:03:03
Contratos_Rel_EX	LYT_CONTRATO_LIMITE	EX	Ok	2018-11-07 10:59:00	2018-11-07 11:02:12
Contratos_Caract_CW	LYT_CARACTERISTICAS	CW	Ok	2018-11-07 11:03:45	2018-11-07 11:03:59
Contratos_Rel_CW	LYT_REL_CONTRATO_PRODUCTO	CW	Ok	2018-11-07 11:05:02	2018-11-07 11:10:02
Contratos_Rel_CW	LYT_REL_CONTRATO_ESTADO	CW	Ok	2018-11-07 11:05:02	2018-11-07 11:07:48
Contratos_Rel_CW	LYT_REL_PERSONA_CONTRATO	CW	Ok	2018-11-07 11:05:02	2018-11-07 11:07:08
Contratos_Rel_CW	LYT_REL_CONTRATO_CARACTERISTICAS	CW	Ok	2018-11-07 11:05:03	2018-11-07 11:10:08
Contratos_Rel_CW	LYT_CONTRATO_LIMITE	CW	Ok	2018-11-07 11:05:03	2018-11-07 11:07:18
Transacciones_EX	LYT_TRX_TARJETA_CREDITO	EX	Ok	2018-11-07 11:10:49	2018-11-07 11:10:55
Transacciones_EX	LYT_COMERCIO	EX	Ok	2018-11-07 11:10:50	2018-11-07 11:11:03
Transacciones_Comercio_CW	LYT_COMERCIO	CW	Ok	2018-11-07 11:12:11	2018-11-07 11:12:43
Transacciones_CW	LYT_TRX_TARJETA_CREDITO	CW	Ok	2018-11-07 11:13:29	2018-11-07 11:15:38
Ubigeo_TC_CW	FND_FACT_TC_LGG_UBIGEO	CW	Ok	2018-11-07 11:16:17	2018-11-07 11:16:27
Fact_Dia_Saldos_CS	FND_FACT_AGG_DAILY_BALANCE	CW	Ok	2018-11-07 11:17:02	2018-11-07 11:27:18
Fact_Mes_Saldos_CS	FND_FACT_AGG_BALANCE_REPORT_MONTH	CW	Ok	2018-11-07 11:28:23	2018-11-07 11:29:25
Fact_Presupuesto_saldo_CS	FND_FACT_BALANCE_BUDGET	CW	Ok	2018-11-07 11:30:16	2018-11-07 11:35:38
Fact_Transacciones_CS	LYT_FACT_TRANSACCION_TC	CW	Ok	2018-11-07 11:36:34	2018-11-07 11:39:57
Fact_Trx_Parque_CS	LYT_FACT_AGMT_ACTIVE_INACTIVE	CW	Ok	2018-11-07 11:40:53	2018-11-07 11:56:51
Fact_Trx_Parque_CS	LYT_FACT_AGG_TRANSACCION_TC	CW	Ok	2018-11-07 11:40:53	2018-11-07 11:41:12
Fact_DISP_CASH_AGG	LYT_REP_DISP_CASH_AGG	CW	Ok	2018-11-07 11:57:39	2018-11-07 11:57:45
Fact_DISP_CASH_PPTO	LYT_REP_DISP_CASH_PPTO	CW	Ok	2018-11-07 11:58:26	2018-11-07 11:58:35
Fact_DISP_CASH_PROY	LYT_REP_DISP_CASH_PROY	CW	Ok	2018-11-07 11:59:16	2018-11-07 11:59:22

**Figura 21 - Detalle de ejecución de cada proceso**

**Fuente: Elaboración propia**

En caso se genere un error, el log de ejecución indicará la ruta donde se encuentra el detalle:

Layout	Modulo_Layout	Estado	FechaHoraInicio	FechaHoraFin
LYT_CONTRATO_SALDO	EX	Ok	13/11/2018 19:32:43	13/11/2018 19:33:05
LYT_CONTRATOS	EX	Error: /Staging/CRM/Logs/BTEQ_LYT_CONTRATOS-13112018183240.log	13/11/2018 19:32:43	13/11/2018 19:33:14
LYT_PERSONA	EX	Ok	13/11/2018 19:32:43	13/11/2018 19:33:48
LYT_CONTRATO_SAVING_HIST	CW	Ok	13/11/2018 19:31:56	13/11/2018 19:31:59
FND_REEMCAN	EX	Ok	13/11/2018 19:31:11	13/11/2018 19:31:12
FND_EMPLEADO	EX	Ok	13/11/2018 19:31:11	13/11/2018 19:31:14
CRM_SRC_ADQ_CLIENTES	CL	Ok	13/11/2018 19:14:58	13/11/2018 19:15:32
CRM_SRC_FFLL_TIENDAS	CL	Ok	13/11/2018 19:14:57	13/11/2018 19:15:35
CRM_SRC_Dotacion_EJECUTIVO	CL	Ok	13/11/2018 19:14:57	13/11/2018 19:15:32
CRM_SRC_UNIDAD_ORG	CL	Ok	13/11/2018 19:14:57	13/11/2018 19:15:27
CRM_SRC_DOTACION_BCA_TEL	CL	Ok	13/11/2018 19:14:57	13/11/2018 19:15:34
CRM_SRC_DOTACION_FFVV_PLANILLA	CL	Ok	13/11/2018 19:14:57	13/11/2018 19:15:31
CRM_SRC_ABP_CONTACT	CL	Ok	13/11/2018 19:14:57	13/11/2018 19:15:31

**Figura 22 – Log de error de ejecución** Fuente: Elaboración propia

## Anexo 8. Malla que controla la secuencia de ejecución

Para la malla se tienen se necesita hacer 2 procesos:

1. Configuración de tabla de procesos y agrupaciones en una tabla de base de datos (IBK\_CONTROL\_MALLA).

Los procesos que se configuran deben tener lo siguiente:

- Nombre del proceso.
- Grupo al que pertenece el proceso.
- Módulo en donde se ejecuta el proceso.
- Proceso predecesor.
- Módulo del proceso predecesor.

El objeto que almacena esta información es la tabla:

```
CREATE MULTiset TABLE E_DW_CONTROL.IBK_CONTROL_MALLA ,NO FALLBACK ,
NO BEFORE JOURNAL,
NO AFTER JOURNAL,
CHECKSUM = DEFAULT,
DEFAULT MERGEBLOCKRATIO
(
Malla VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC,
Grupo VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC,
Layout VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC,
Modulo_Layout VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC,
Predecesor VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC,
Modulo_Predecesor VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC,
EjecAnterior VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC,
Bloqueante VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC
)
PRIMARY INDEX (Grupo,Layout);
```

A continuación se muestra los procesos configurados para un proyecto donde se utilizó el nuevo framework:

Tabla 13: Configuración de malla de procesos.

Grupo	Layout	Módulo Layout	Predecesor	Modulo Predecesor
Contratos_CL	CRM_SRC_APERTURAS_ACT	CL		
Contratos_CL	CRM_SRC_APERTURA_PANAGO	CL		
Contratos_EX	LYT_CONTRATOS	EX	CRM_SRC_APERTURAS_ACT	CL
Contratos_EX	LYT_CONTRATOS	EX	CRM_SRC_APERTURA_PANAGON	CL
Contratos_EX	LYT_CONTRATOS	EX	FND_CAPTACIONES_SALDOS	CW
Contratos_EX	LYT_CONTRATOS	EX	FND_REEMCAN	EX
Contratos_EX	LYT_REL_CONTRATO_CANAL	EX	CRM_SRC_APERTURA_PANAGON	CL
Contratos_EX	LYT_REL_CONTRATO_CANAL	EX	FND_CAPTACIONES_SALDOS	CW
Contratos_EX	LYT_REL_PER_CONTRATO	EX	CRM_SRC_APERTURA_PANAGON	CL
Contratos_EX	LYT_REL_PER_CONTRATO	EX	CRM_SRC_APERTURAS_ACT	CL
Contratos_EX	LYT_REL_PER_CONTRATO	EX	FND_CAPTACIONES_SALDOS	CW
Contratos_EX	LYT_CONTRATO_PRODUCTO	EX	CRM_SRC_APERTURA_PANAGON	CL
Contratos_EX	LYT_CONTRATO_PRODUCTO	EX	FND_CAPTACIONES_SALDOS	CW
Dotacion_CL	CRM_DOTACION_BCA_TEL	CL		
Dotacion_CL	CRM_SRC_Dotacion_EJECUTIVO	CL		
Dotacion_CL	CRM_SRC_ABP_CONTACT	CL		
Dotacion_CL	CRM_SRC_ADQ_CLIENTES	CL		
Dotacion_CL	CRM_SRC_FLL_TIENDAS	CL		
Dotacion_CL	CRM_SRC_UNIDAD_ORG	CL		
Empleado_EX	FND_REEMCAN	EX	CRM_DOTACION_BCA_TEL	CL
Empleado_EX	FND_REEMCAN	EX	CRM_SRC_Dotacion_EJECUTIVO	CL

Empleado_EX	FND_REEMCAN	EX	CRM_SRC_ABP_CONTACT	CL
Empleado_EX	FND_REEMCAN	EX	CRM_SRC_ADQ_CLIENTES	CL
Empleado_EX	FND_REEMCAN	EX	CRM_SRC_FFLI_TIENDAS	CL
Empleado_EX	FND_REEMCAN	EX	CRM_SRC_UNIDAD_ORG	CL
Empleado_EX	FND_EMPLEADO	EX	CRM_DOTACION_BCA_TEL	CL
Empleado_EX	FND_EMPLEADO	EX	CRM_SRC_Dotacion_EJECUTIVO	CL
Empleado_EX	FND_EMPLEADO	EX	CRM_SRC_ABP_CONTACT	CL
Empleado_EX	FND_EMPLEADO	EX	CRM_SRC_ADQ_CLIENTES	CL
Empleado_EX	FND_EMPLEADO	EX	CRM_SRC_FFLI_TIENDAS	CL
Empleado_EX	FND_EMPLEADO	EX	CRM_SRC_UNIDAD_ORG	CL
Geografia_EX	FND_ELEMGEOG	EX	CRM_SRC_MAESTRO_TIENDAS	CL
Geografia_EX	FND_RECANEGE	EX	CRM_SRC_MAESTRO_TIENDAS	CL
Geografia_CW	FND_ELEMGEOG	CW	FND_ELEMGEOG	EX
Geografia_Rel_CW	FND_RECANEGE	CW	FND_ELEMGEOG	CW
Sitio_CL	CRM_SRC_MAESTRO_TIENDAS	CL		
Channel_CW	FND_CHANNEL_INSTANCE	CW	LYT_CONTRATOS	EX
Contratos_CW	LYT_CONTRATOS	CW	LYT_CONTRATOS	EX
Rel_EmCanal_CW	FND_RELEMPCANALCAPTAC	CW	CRM_SRC_APERTURA_PANAGON	CL
Rel_EmCanal_CW	FND_RELEMPCANALCAPTAC	CW	CRM_SRC_APERTURAS_ACT	CL
Rel_EmCanal_CW	FND_RELEMPCANALCAPTAC	CW	FND_REEMCAN	EX
Rel_EmCanal_CW	FND_RELEMPCANALCAPTAC	CW	LYT_CONTRATOS	EX
Contratos_CW	LYT_CONTRATOS	CW	FND_RELEMPCANALCAPTAC	CW
Contratos_Rel_CW	LYT_REL_CONTRATO_CANAL	CW	LYT_CONTRATOS	CW
Contratos_Rel_CW	LYT_REL_PER_CONTRATO	CW	LYT_CONTRATOS	CW
Contratos_Rel_CW	LYT_CONTRATO_PRODUCTO	CW	LYT_CONTRATOS	CW
Contratos_CW	LYT_CONTRATOS	CW	FND_CAPTACIONES_SALDOS	CW



Contratos_CW	LYT_CONTRATOS	CW	FND_CHANNEL_INSTANCE	CW
Contratos_CW	LYT_CONTRATOS	CW	FND_LYT_EMPLEADO	CW
Empleado_CW	FND_LYT_EMPLEADO	CW	FND_EMPLEADO	EX
Capt_Saldos_CW	FND_CAPTACIONES_SALDOS	CW		
Saldos_APR_CW	FND_SALDOS_APR	CW	FND_AGRE_UPDCAP	CW
Saldos_APRCW	FND_AGRE_UPDCAP	CW	LYT_CONTRATOS	CW

Fuente: Elaboración propia

Los procesos configurados se utilizan para ejecutar de manera integral los procesos correspondientes que se encuentren relacionados:

#-----Contratos\_CL-----

/Staging/CRM/Shell/Sh\_Malla\_IBK.sh Captaciones Contratos\_CL 20180501

sh /Staging/CRM/Shell/ejecCargaArchivo.sh Apertura

CRM\_SRC\_APERTURA\_PANAGON 20180312

sh /Staging/CRM/Shell/ejecCargaArchivo.sh Aperturas\_ACT

CRM\_SRC\_APERTURAS\_ACT 20180419

/Staging/CRM/Shell/Sh\_Malla\_IBK.sh Captaciones Empleado\_EX

sh /Staging/CRM/Shell/Sh\_Genera\_Layout.sh FND\_EMPLEADO

sh /Staging/CRM/Shell/Sh\_Genera\_Layout.sh FND\_REEMCAN

#---- CAPTACIONES SALDOS

/Staging/CRM/Shell/Sh\_Malla\_IBK.sh Captaciones Captaciones\_Saldos\_CW

sh /Staging/CRM/Shell/Sh\_EjecutaProcedure\_FDM.sh

FND\_CAPTACIONES\_SALDOS D\_DW\_STAGING

2. Shells de configuración se la secuencia de ejecución.

Para definir la secuencia en que se ejecutarán los procesos que se hayan configurado se implementaron las Shell:

- La Shell Sh\_Malla\_IBK.sh que ejecuta el proceso configurado de acuerdo al módulo que pertenezca:

```

1  #!/bin/sh
2  #####
3  # PROYECTO      :
4  # SCRIPT ID     : Ejec
5  # FUNCION       : Lanzar extraccion, carga staging y carga modelo
6  # AUTOR        : DW - Juan Sánchez
7  #####
8
9
10 dattim=`date +%d%m%Y%H%M%S`
11
12
13 #VARIABLES INICIALES
14 PRM_PATH_SHELL=`dirname $0|awk '{if($0==".") print "pwd" ; else print "dirname "'$0'"}'|ksh`
15 . ${PRM_PATH_SHELL}/Include_CRM.sh
16
17 NOMARCHIVO=${GRUPO$dattim}
18
19 MALLA=${1}
20 GRUPO=${2}
21 NOMARCHIVOPID=${3}
22 FECHA=${4}
23
24 V_PRM_PATH_LOG=${PRM_PATH_LOG}
25 TEMPCOMANDO=${GRUPO} tmo.txt"

```

---

```

129
130 fEjecutaProceso()
131 {
132     filename=${PRM_PATH_SHELL}/${GRUPO$dattim}_final.txt"
133     echo $filename
134
135     while read -r line
136     do
137         LineaLote="$line"
138         EjecShellCarga=${LineaLote}
139         echo "Comando: $EjecShellCarga"
140         sh "$EjecShellCarga"
141     done < "$filename"
142
143     rm -f ${PRM_PATH_SHELL}/${GRUPO$dattim}_final.txt"
144 }
145
146 fConsolidaPredecesor()
147 {
148     filename=${PRM_PATH_SHELL}/${GRUPO$dattim}_comando.txt"
149     echo $filename
150
151     while read -r line
152     do
153         LineaLote="$line"
154         ProcesoShell=${ProcesoShell} - "$LineaLote
155

```

**Figura 23 – Shell de malla de ejecución de procesos      Fuente: Elaboración propia**

- La Shell Sh\_Malla\_seq.sh que los procesos en la secuencia que se requiere, esta Shell invoca a Sh\_Malla\_IBK.sh.

```

10 dattim=`date +%d%m%Y%H%M%S`
11
12
13 #VARIABLES INICIALES
14 PRM_PATH_SHELL=`dirname $0|awk '{if($0==".") print "pwd" ; else print "dirname "'$0'"}'|ksh`
15 . ${PRM_PATH_SHELL}/Include_CRM.sh
16
17 MALLA=$1
18 FECHA=$2
19 periodo=`expr substr $2 1 6`
20
21 V_PRM_PATH_LOG=${PRM_PATH_LOG}
22 NOMARCHIVOPID="PID_${MALLA}_${dattim}.pid"
23 CANTARCHIVOPID="PID_${MALLA}_${dattim}.txt"
24
25 fValidaPredecesor()
26 {
27     if test ! -f ${PRM_PATH_SHELL}/${NOMARCHIVOPID}
28     then
29         echo "001 : Error"
30         exit 1
31     else
32         echo "Continúa ejecución"
33     fi
34 }
35
36 fValidaStatus()
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59 echo "Captaciones Saldos"
60 sh ${PRM_PATH_SHELL}/Sh_Malla_IBK.sh $MALLA Captaciones_Saldos_CW "$NOMARCHIVOPID"
61 sleep 15
62 fValidaPID
63
64
65
66
67
68
69
70
71
72
73
74 echo "Carga Archivos Dotación"
75 sh ${PRM_PATH_SHELL}/Sh_Malla_IBK.sh $MALLA Dotacion_CL "$NOMARCHIVOPID" ${periodo}
76 sleep 15
77 fValidaPID
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

**Figura 24 – Shell de secuencia de ejecución** Fuente: Elaboración propia

En esta Shell se define la siguiente secuencia de ejecución:

- Dotacion: Carga a Landing
- Contratos\_CL: Carga a Landing
- Saldos\_APR\_CAP: Carga a Landing
- Empleado\_EX: Carga a Layout
- Contrato\_EX: Carga a Layout
- Empleado\_CW: Carga a FDM
- Relacion\_EmCanal\_CW: Carga a FDM
- Contratos\_Rel\_CW: Carga a FDM
- Saldos\_UPD: Carga a FDM
- Saldos\_APR: Carga a FDM
- Saldos\_FACT: Carga a FDM



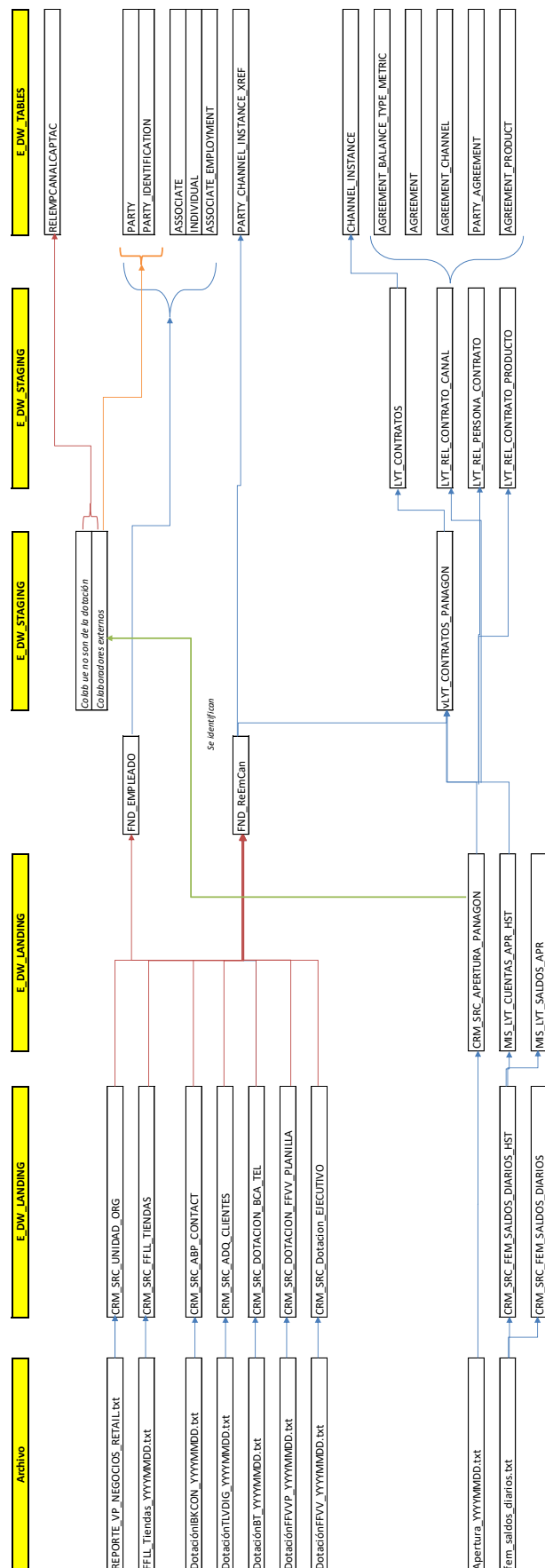


Figura 25 - Malla de procesos Fuente: Elaboración propia

En caso se requiera validar la ejecución de los procesos desde base de datos, se debe consultar la tabla de control, para el ejemplo presentado se tiene los siguientes queries de validación:

```
SELECT * FROM E_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
WHERE LAYOUT IN
(
'CRM_DOTACION_BCA_TEL',
'CRM_SRC_DOTACION_FFVV_PLANILLA',
'CRM_SRC_Dotacion_EJECUTIVO',
'CRM_SRC_ABP_CONTACT',
'CRM_SRC_ADQ_CLIENTES',
'CRM_SRC_FFLL_TIENDAS',
'CRM_SRC_UNIDAD_ORG'
);
```

```
SELECT * FROM E_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
WHERE LAYOUT='CRM_SRC_APERTURA_PANAGON';
```

```
SELECT * FROM E_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
WHERE LAYOUT IN
(
'FND_REEMCAN',
'FND_EMPLEADO'
);
```

```
SELECT * FROM E_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
WHERE LAYOUT='FND_LYT_EMPLEADO';
```

```
SELECT * FROM E_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
WHERE LAYOUT='LYT_CONTRATOS';
```

```
SELECT * FROM E_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
WHERE LAYOUT='FND_CHANNEL_INSTANCE';
```

```
SELECT * FROM E_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
WHERE LAYOUT='FND_RELEMPCANALCAPTAC';
```

```
SELECT * FROM E_DW_CONTROL.DBC_LOGEJECUTAPROCESOETL
WHERE LAYOUT='LYT_CONTRATOS';
```

**SELECT \* FROM E\_DW\_CONTROL.DBC\_LOGEJECUTAPROCESOETL**  
**WHERE LAYOUT='FND\_AGRE\_UPDCAP';**

**SELECT \* FROM E\_DW\_CONTROL.DBC\_LOGEJECUTAPROCESOETL**  
**WHERE LAYOUT='FND\_SALDOS\_APR';**

**SELECT \* FROM E\_DW\_CONTROL.DBC\_LOGEJECUTAPROCESOETL**  
**WHERE LAYOUT='FND\_FACT\_BALANCE\_COMMONCOA';**

### Anexo 9. Reportes de control de procesos

Los reportes desarrollados en Power BI facilitarán al área de operaciones el monitoreo de los procesos del área de Data Analytics:

- Reporte de estado de ejecución de los procesos.

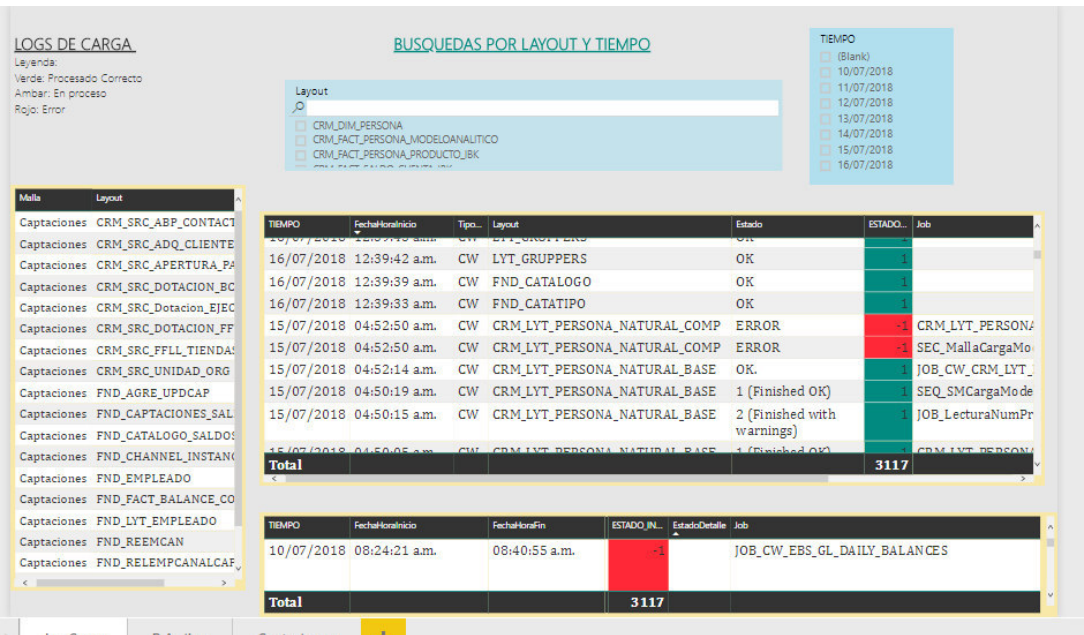


Figura 26 - Reporte de control Fuente: Elaboración propia

- Reporte de fechas de información cargada en el datawarehouse



Figura 27 - Reporte de información cargada Fuente: Elaboración propia

- Reporte de fechas y montos de un proceso cargado.

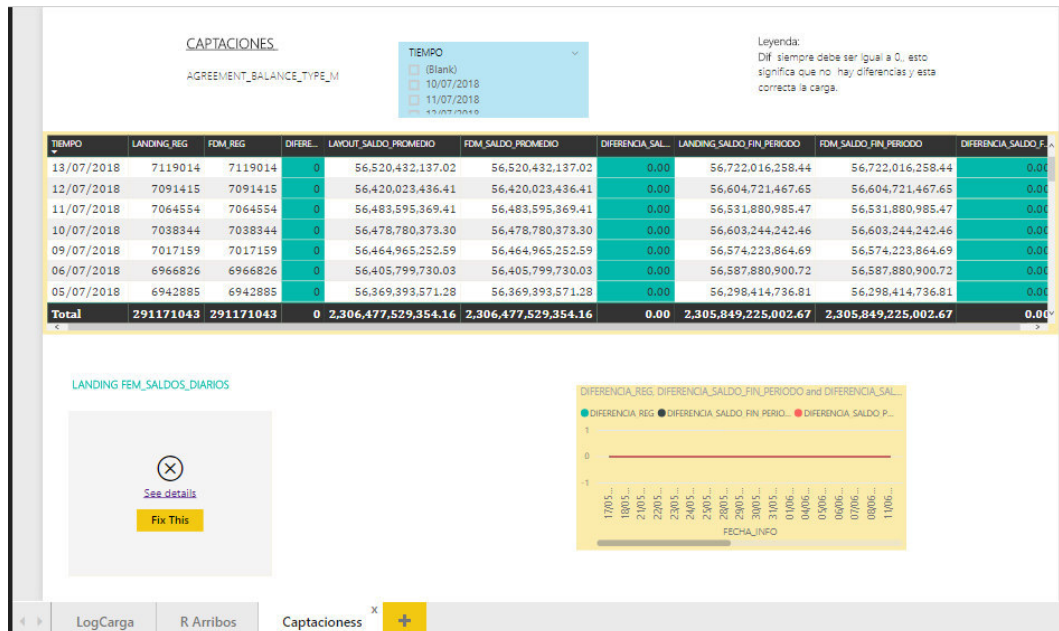
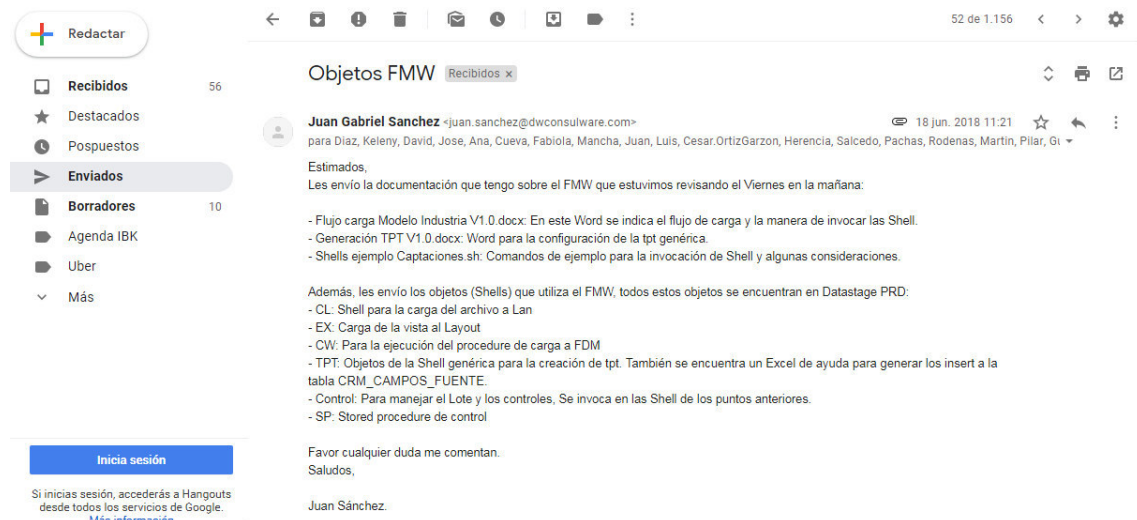


Figura 28 - Reporte de fechas procesadas Fuente: Elaboración propia

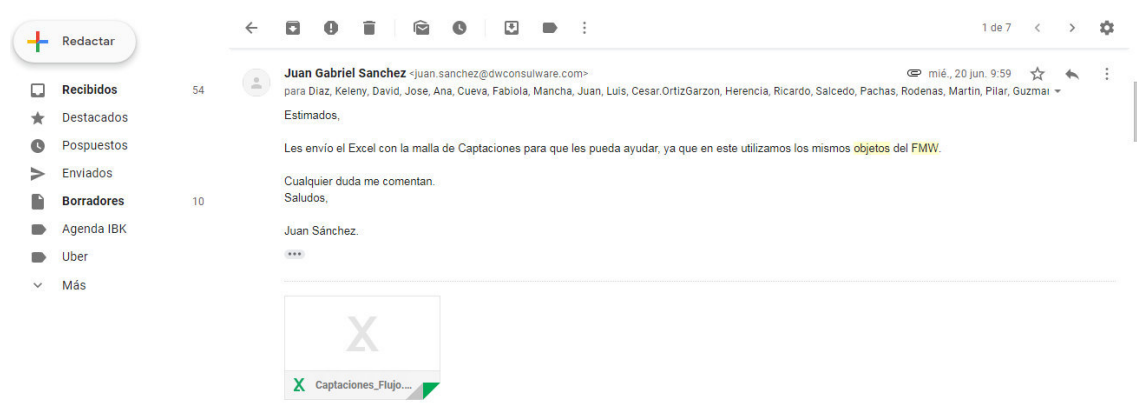
## Anexo 10. Correo de difusión de objetos

Luego de haberse hecho la capacitación en el nuevo framework, se envió correos para conocimiento de los involucrados sobre los componentes del framework.

- Correo donde se da a conocer a los involucrados, los componentes que se han expuesto en este documento:



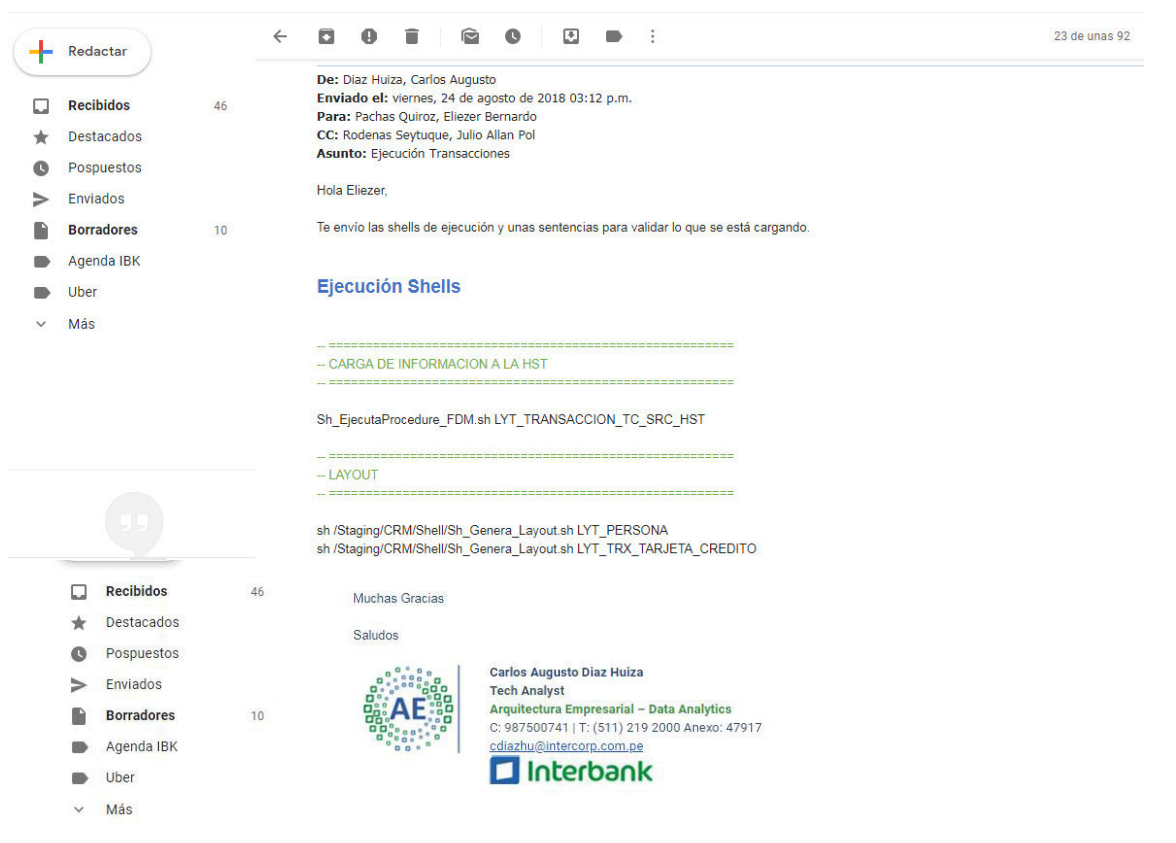
- Correo donde se transfiere una malla donde se utiliza los objetos del framework implementado:



- Correo de transferencia de la configuración de la malla, detallada en el presente documento:



- Correo donde se observa que los objetos del framework están siendo utilizados por las personas involucradas del área de Data Analytics:



- Redactor

Recibidos 46

Destacados

Pospuestos

Enviados

Borradores 10

Agenda IBK

Uber

Más

Inicia sesión

Si inicias sesión, accederás a Hangouts desde todos los servicios de Google.  
Más información

De: Mancha Paredes, Yolanda Cecilia [mailto:[Yolanda.ManchaParedes@Teradata.com](mailto:Yolanda.ManchaParedes@Teradata.com)]

Enviado el: viernes, 24 de agosto de 2018 02:40 p.m.

Para: Pachas Quiroz, Elezer Bernardo

CC: Herencia Mondragon, Manuel; Ofuentes, Antonio

Asunto: [ IBK ] - [ TC ] - Carga Historica de informacion

Importancia: Alta

Elezer,

Según lo conversado, la ejecución se realiza del bloque completo a partir del 30-05-2018.

Los pasos a seguir son:

Copiar el .ksh en la ruta /Staging/CRM/Shell/ y asignarle permisos

Modificar las credenciales: User y password

Desde el putty ubicarse en la siguiente ruta : cd /Staging/CRM/Shell/ y los comandos de ejecución son los siguientes:

Cabe resaltar, que la ejecución aplica para todo el bloque de Contratos TC Relacional

```
## CONTRATOS
sh /Staging/CRM/Shell/Sh_Genera_Layout.sh LYT_CONTRATOS; if [ $? -ne 0 ]; then exit 1 ; fi
sh /Staging/CRM/Shell/Sh_EjecutaProcedure_FDM.sh LYT_CONTRATOS E_DW_STAGING; if [ $? -ne 0 ]; then exit 1 ; fi

# Producto
sh /Staging/CRM/Shell/Sh_Genera_Layout.sh LYT_REL_CONTRATO_PRODUCTO; if [ $? -ne 0 ]; then exit 1 ; fi
sh /Staging/CRM/Shell/Sh_EjecutaProcedure_FDM.sh LYT_REL_CONTRATO_PRODUCTO E_DW_STAGING; if [ $? -ne 0 ]; then exit 1 ; fi

# Características
sh ejecCargaArchivo.sh CondicionEconomica FND_SRC_CONDICION ECONOMICA '$(PRM_FECHA_PROCESO)'; if [ $? -ne 0 ]; then exit 1 ; fi
sh /Staging/CRM/Shell/Sh_Genera_Layout.sh LYT_CARACTERISTICAS; if [ $? -ne 0 ]; then exit 1 ; fi
sh /Staging/CRM/Shell/Sh_EjecutaProcedure_FDM.sh LYT_CARACTERISTICAS E_DW_STAGING; if [ $? -ne 0 ]; then exit 1 ; fi

# Contrato con Características
sh /Staging/CRM/Shell/Sh_Genera_Layout.sh LYT_REL_CONTRATO_CARACTERISTICAS; if [ $? -ne 0 ]; then exit 1 ; fi
sh /Staging/CRM/Shell/Sh_EjecutaProcedure_FDM.sh LYT_REL_CONTRATO_CARACTERISTICAS E_DW_STAGING; if [ $? -ne 0 ]; then exit 1 ; fi

# Contrato con Limites
sh /Staging/CRM/Shell/Sh_Genera_Layout.sh LYT_CONTRATO_LIMITE; if [ $? -ne 0 ]; then exit 1 ; fi
sh /Staging/CRM/Shell/Sh_EjecutaProcedure_FDM.sh LYT_CONTRATO_LIMITE E_DW_STAGING; if [ $? -ne 0 ]; then exit 1 ; fi

### Persona Contrato
sh /Staging/CRM/Shell/Sh_Genera_Layout.sh LYT_REL_PERSONA_CONTRATO; if [ $? -ne 0 ]; then exit 1 ; fi
sh /Staging/CRM/Shell/Sh_EjecutaProcedure_FDM.sh LYT_REL_PERSONA_CONTRATO E_DW_STAGING; if [ $? -ne 0 ]; then exit 1 ; fi

# Contrato Estados
sh /Staging/CRM/Shell/Sh_Genera_Layout.sh LYT_REL_CONTRATO_ESTADO; if [ $? -ne 0 ]; then exit 1 ; fi
sh /Staging/CRM/Shell/Sh_EjecutaProcedure_FDM.sh LYT_REL_CONTRATO_ESTADO E_DW_STAGING; if [ $? -ne 0 ]; then exit 1 ; fi
```

Cualquier consulta con gusto.

Saludos,

Yolanda Mancha  
PS Consultant

TERADATA.

Cruceación del Cule Golf Las Incas 154  
Of. 701 - Edificio Capital Golf, Surco, Lima  
[yolanda.mancha@teradata.com](mailto:yolanda.mancha@teradata.com)  
[teradata.com](http://teradata.com)